
MARZO 1 9 8 8
REVISTA PARA USUARIOS DE LA NORMA
LOAD
MSX
AÑO 2 Nº 23
A 9 REP.ARGENTINA
COMPUJUEGOS
¿SIRVEN PARA ALGO?
COMO REDUCIR
LOS PROGRAMAS
INEDITOS
• MEMOREX
• CONTROL DE
INGRESOS
• PARA REVISAR
EL DISQUETE
CONCURSO
Y SORTEO

A la computadora personal
Talent MSX
nada le es imposible

dialogo - D. P.



MSX es marca registrada de MICROSOFT CORPORATION

Porque gracias a la norma internacional MSX, la TALENT MSX trasciende sus propios límites.
Hasta ahora, cuando usted compraba una computadora personal de cualquier marca, quedaba automáticamente desconectado del resto del mundo de la computación. Porque los distintos equipos y sistemas no eran compatibles entre sí.
Hasta que dos grandes empresas de informática, la Microsoft Corp. de EE.UU. y la ASCII del Japón se pusieron de acuerdo para crear una norma standard: la MSX. Que se expandió también rápidamente en Europa. Y que hoy TALENT presenta por primera vez en la Argentina.
Mientras que la mayoría de las computadoras de su tipo que se ofrecen en el mercado nacional, han sido discontinuadas por obsoletas en sus lugares de origen, TALENT MSX tiene casi ilimitadas posibilidades de desarrollo. Porque la norma MSX es en todo el mundo inteligencia en crecimiento.
La TALENT MSX pone a su disposición un mundo de software para elegir. Y con la incorporación de todos sus periféricos llega a ser una auténtica computadora profesional.

UTIL

Su poderoso sistema operativo MSX permite el acceso a todo tipo de procesamiento de datos:

- Planillas de cálculo.
- Procesadores de palabra.
- Gráficos de negocios.
- Bases de datos (d Base II, etc.)
- Contabilidad general, sueldos, y jornales, costos, etc., desarrollados bajo CP/M en Basic, Cobol, Pascal o C.

Con la posibilidad de conexión a línea telefónica permite la transferencia y consulta de datos entre computadoras personales, profesionales o bancos de datos.
La grabación de archivos es en formato MS-DOS, haciéndola compatible con las computadoras profesionales.

DIDACTICA

Dispone de tres lenguajes para la enseñanza de computación: LOGO como lenguaje de inducción para los mas chicos. Lenguaje de Programación en castellano, para todos los que quieran aprender a programar sin conocimientos previos. Y Basic MSX como lenguaje profesional. Más una amplia variedad de periféricos como el Mouse, Lápiz Optico, Tableta gráfica, Track-ball, etc.

DIVERTIDA

La más genial para Video-Juegos. Por la amplísima biblioteca de programas **–todos nuevos–** de la norma MSX en el mundo. Y además, el Basic MSX permite al usuario generar sus propios juegos con un manejo tan simple, como sólo TALENT MSX puede ofrecer.

CARACTERISTICAS TECNICAS

- Memoria principal 64 KB ampliable hasta 576 KB.
- Memoria de video: 16 KB RAM.
- ROM incorporada de 32 KB con el MSX-Basic de Microsoft.
- Gráficos completos, hasta 32 sprites y 16 colores simultáneos.
- Generador de sonido de 3 voces y 8 octavas.
- Conexión para cualquier grabador.
- Interfaz para salida impresora paralela.
- Conectores para cartuchos y expansiones.
- Fuente para 220 V y modulador PAL-N incorporado.

DISTRIBUIDORES AUTORIZADOS: CAPITAL FEDERAL: AMATRIX, Bolívar 173 - ARGECINT, Av. de Mayo 1402 - BAIDAT COMPUTACION, Juramento 2349 - COMPUPRANDO, Av. de Mayo 965 - COMPUSHOP, Córdoba 1464 - COMPUTIQUE, Córdoba 1111, E. P. - COMPUTRONIC, Viamonte 2096 - CP67 CLUB, Florida 683, L. 18 - DALTON COMPUTACION, Cabildo 2283 - ELAB, Cabildo 730 - MICROSTAR, Callao 462 - Q.S.P., Bartolomé Mitre 864 - SERVICIOS EN INFORMATICA, Paraná 164 - DISTRIBUIDORA CONCALES, Tucumán 1458 - MICROMATICA, Av. Pueyrredón 1135 - **ACASSUSO:** MICROSTAR ACASSUSO, Eduardo Costa 892 - **AVELLANEDA:** ARGOS, Av. Mitre 1755 - **BOULOGNE:** COMPUTIQUE CARREFOUR, Bernardo de Irigoyen 2647 - **CASTELAR:** HOT BIT COMPUTACION, Carlos Casares 997 - **LANUS:** COMPUTACION LANUS, Caaguazú 2186 - **LOMAS DE ZAMORA:** ARGESIS COMPUTACION, Av. Meeks 269 - **MARTINEZ:** VIDEO BYTE, Hipólito Yrigoyen 32 - **RAMOS MEJIA:** MANIAC COMPUTACION, Rivadavia 13734 - **SAN ISIDRO:** FERNANDO CORATELLA, Cosme Beccar 249 - **VICENTE LOPEZ:** SERVICIOS EN INFORMATICA, Av. del Libertador 882 - **BAHIA BLANCA:** SERCOM, Donado 327 - SUMASUR, Alsina 236 - **LA PLATA:** CADEMA, Calle 7 Nº 1240 - CERO-UNO INFORMATICA, Calle 48 Nº 529 - **MAR DEL PLATA:** FAST, Catamarca 1755 - **NECOCHEA:** CAFAL, Calle 57 Nº 2920 - SERCOM, Calle 57 Nº 2216 - **TRENQUE LAUQUEN:** COMPUQUEN, Villegas 231 - **CORDOBA:** AUTODATA, Pasaje Santa Catalina 27 - TECSIEM, Santa Rosa 715 - **ROSARIO:** 2001 COMPUTACION, Santa Fe 1468 - MINICOMP, Maipú 862 - SISOR, Urquiza 1062 - **SANTA FE:** ARGECINT, P. San Martín 2433, L. 36 - SISOR, Rivadavia 2553 - INFORMATICA, San Gerónimo 2721/25 - **VILLA MARIA:** JUAN CARLOS TRENTO, 9 de Julio 80 - **MENDOZA:** INTERFACE, Sarmiento 98 - BIT & BYTE, 9 de Julio 1030 - **COMODORO RIVADAVIA:** COMPUSER, 25 de Mayo 827 - **GENERAL ROCA:** DISTRIBUIDORA VECCHI, 25 de Mayo 762 - **LA PAMPA:** MARINELLI, Pellegrini 155 - **NEUQUEN:** MEGA, Perito Moreno 383 - EDISA, Roca esq. Fotheringham - **RIO GRANDE:** INFORMATICA M & B, Perito Moreno 290 - **SAN CARLOS DE BARILOCHE:** L. ROBLEDO & ASOCIADOS, Elfein 13, Piso 1º - **TRELEW:** SISTENOVA, Sarmiento 456 - **PARANA:** MARIO GARCIA, Laprida y Santa Fe - **POSADAS:** CENTRO DE COMPUTOS ELDORADO, Colón 2429 - **RESISTENCIA:** FRANCO SANTI, Carlos Pellegrini 761 - **SAN**

STAFF

LOAD MSX

Director Periodístico

Fernando Flores

Secretario de Redacción

Ariel Testori

Redacción

Fernando Pedró

Arte y Diagramación

Fernando Amengual y
Tamara Migelson

Departamento de Avisos

Oscar Devoto y
Nelzo Capello

Departamento de Publicidad

Dolores Urien

Servicios Fotográficos

Image Bank, Oscar Burriel y
Victor Grubicy

EDITORIAL PROEDI

Presidente

Ernesto del Castillo

Vicepresidente

Cristian Pusso

Director Titular

Javier Campos Malbrán

Director Suplente

Armengol Torres Sabaté

Load Revista para usuarios de la norma MSX es una publicación mensual editada por Editorial PROEDI S.A., Paraná 720, 5° Piso, (1017) Buenos Aires. Tel.: 46-2886 y 49-7130. Radiollamada: Tel.: 311-0056 y 312-6383, código 5941. Registro Nacional de la Propiedad Intelectual: E. T. M. Registrada. Queda hecho el depósito que indica la Ley 11.723 de la Propiedad Intelectual.

ISSN 0326-8241

Impreso en Impresiones Gráficas Tabaré S.A.I.C. Erézcano 3158 Cap. - Fotocromo tapa: Columbia. Fotocomposión: Interamericana Gráfica.

Los ejemplares atrasados se venderán al precio del último número en circulación.

SUMARIO

EL USUARIO JUGUETON



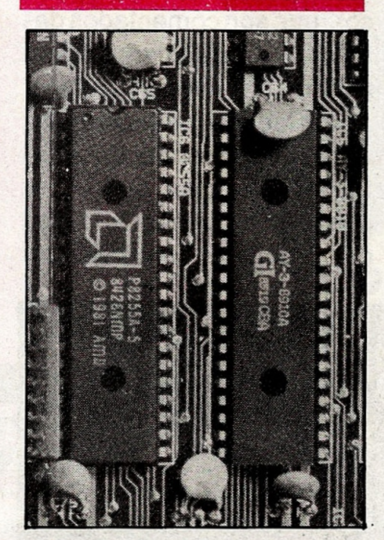
¿Hasta dónde es beneficioso el contacto continuo con la televisión en la educación de un adolescente? Y aún más, ¿cuál es la incidencia de los video juegos con el clásico esquema del "matamarcianos" en la formación de los niños sin iniciativa? (Pág. 6)

INTERPRETES Y COMPILADORES, ¿SON ENEMIGOS?

¿En qué casos se justifica el uso del compilador? ¿Cómo se usa? ¿Qué diferencia hay con el intérprete? Aquí contestamos estos interrogantes tratando de poner luz sobre el tema. (Pág. 10)

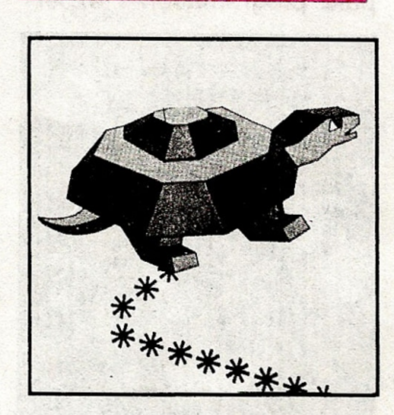
LOS MISTERIOS DEL MSX-DOS:

RINCON DEL USUARIO



En computación, cuando se quiere expresar la idea de "circuito electrónico miniaturizado" simplemente se emplea la palabra "chip" que significa "pastilla", y de esta manera nos vamos entendiendo. (Pág. 20)

ANIMACION LOGO



Los docentes y usuarios de Logo en varias oportunidades necesitan incentivar a sus alumnos y niños respectiv-

notas apunta a aportar información y sugerencias a los lectores que puedan inscribirse en el rubro de usuarios serios, en contraposición con los usuarios lúdicos, sin ánimo peyorativo alguno. (Pág. 28)

CONTROL DE INGRESO DE DATOS

Es muy conveniente armarnos una librería de subrutinas modulares, que nos brinden soluciones a problemas que se presentan habitualmente en la confección de nuestros programas: manejo de interrupciones, errores, fechas, conversores de números a letras, etcétera. (Pág. 14)

PROGRAMAS



Memorex (Pág. 12) - Dump (Pág. 24)

SECCIONES FIJAS

ACAMATICA

La empresa Telemática continúa brindando ayuda técnica al Automóvil Club Argentino (ACA), a través de ACAmática.
Telemática instala los PASA (Puestos de Atención Servicio ACAMATICA) en distintas dependencias de la red del Automóvil Club. Además da capacitación gratuita en el uso de ACAmática a los socios , cualquiera sea la computadora que posean.
Telemática también pone a disposición del ACA la totalidad de los Centros de Asistencia al Usuario. Los centros brindancapacitacióngratuita en el uso de la red mencionada.

LOS SECRETOS DE LA DEPURACION DE SOFTWARE

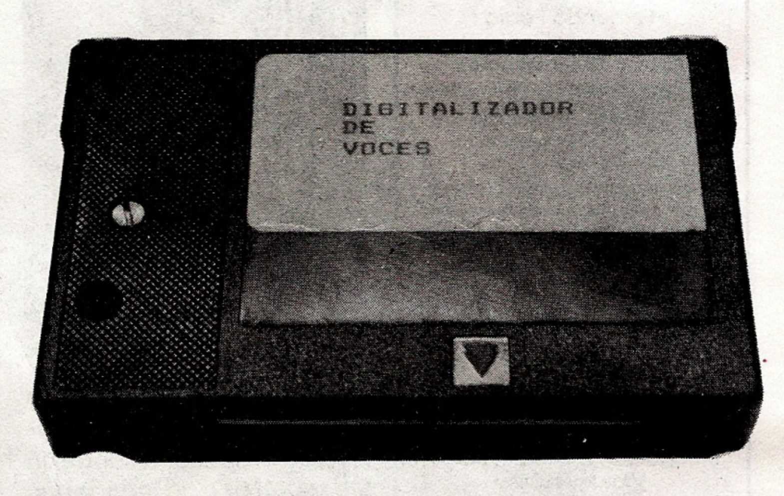
Autor: Truck Smith

El objetivo del libro es enseñar las técnicas de depuración de programas.
Se trata de enseñar a pensar lógicamente para desentrañar un problema, de aprender a aportar la mayor cantidad de conocimientos al programa, hacer pruebas imaginativas y realizar observaciones. Las depuraciones no son sencillas, se asemejan a la resolución de un crucigrama o un rompecabezas.
El proceso de depuración está compuesto por varios pasos, que son comentados detalladamente a lo largo de esta obra. Primero, el planteo y discusión del problema; segundo, suponer dónde está; tercero, suponer qué es lo que no funciona, y luego comprobar las suposiciones, refinar y repetir el proceso hasta encontrar el error, determinar la solución y, por último, hacer las correcciones.
La obra está orientada tanto para programadores exper-



tos como novatos, siempre y cuando conozcan los conceptos básicos del lenguaje de programación en el que están trabajando. Este libro propone ejemplos en BASIC, FORTRAN, PASCAL y lenguaje ensamblador. El programador sin mucha experiencia encontrará aquí una valiosa ayuda y orientación, mientras que el experto tropezará con técnicas y sugerencias para mejorar tal vez su estilo de depuración.(Edita : Edexim).

DIGITALIZADOR DE VOCES



Si bien puede sonar algo complicado, el funcionamiento de un digitalizador de voces es muy sencillo, y su función es clara: permite almacenar hasta un minuto de una conversación, canción o mensaje en la memoria de la máquina, y luego se puede guardar en disquete. Las aplicaciones para este periférico son muchas, y su versatilidad es muy amplia.
La entrada de sonido a la computadora se realiza por medio de la conexión de EAR del grabador.
De esta forma, podemos digitalizar cualquier cosa que esté grabada en un casete, ya sea música, un programa o un mensaje. La salida del mensaje digitalizado se realiza a través del televisor, o de la salida de audio del monitor.
El sistema se suministra en un cartridge que se conecta en el port de las MSX.
Una vez hecho esto, se pone en funcionamiento el equipo y se llama al digitalizador mediante una instrucción USR a una zona de memoria determinada.
A partir de aquí, pasamos al menú principal que nos brinda las siguientes opciones:
1. Cargar datos ya digitalizados.
2. Salvar al disco los datos.
3. Directorio del disquete
4. Leer casete.
5. Emitir voz completa.
6.- Emitir voz por partes.
7. Eliminar el programa de la memoria. Junto con las instrucciones del digitalizador se suministra un programa de ejemplo, que se puede agregar a cualquier otro escrito por nosotros.

El cartucho se alimenta de la máquina, y no hay ningún cable ni conexión externa que realizar. De esta forma, se eliminan las posibilidades de malas conexiones, o cortocircuitos en áreas delicadas de la computadora.
El sistema funciona por medio de un conversor analógico/digital construido por soft, más un sistema que se encarga de convertir estos bloques de bytes a un archivo compatible con el sistema MSX-DOS.
Las posibilidades de controlar una grabación digitalizada por computadora son infinitas, y no faltará algún usuario jugetón que le encuentre la vuelta al sistema para convertir este periférico en un "deformador" de voces, manteniéndolas entendibles pero irreconocibles.
En fin, se trata de un periférico nuevo, que no solo puede servir para jugar.
(Fabricante: COMPUTRONIC)

MSX-TEST

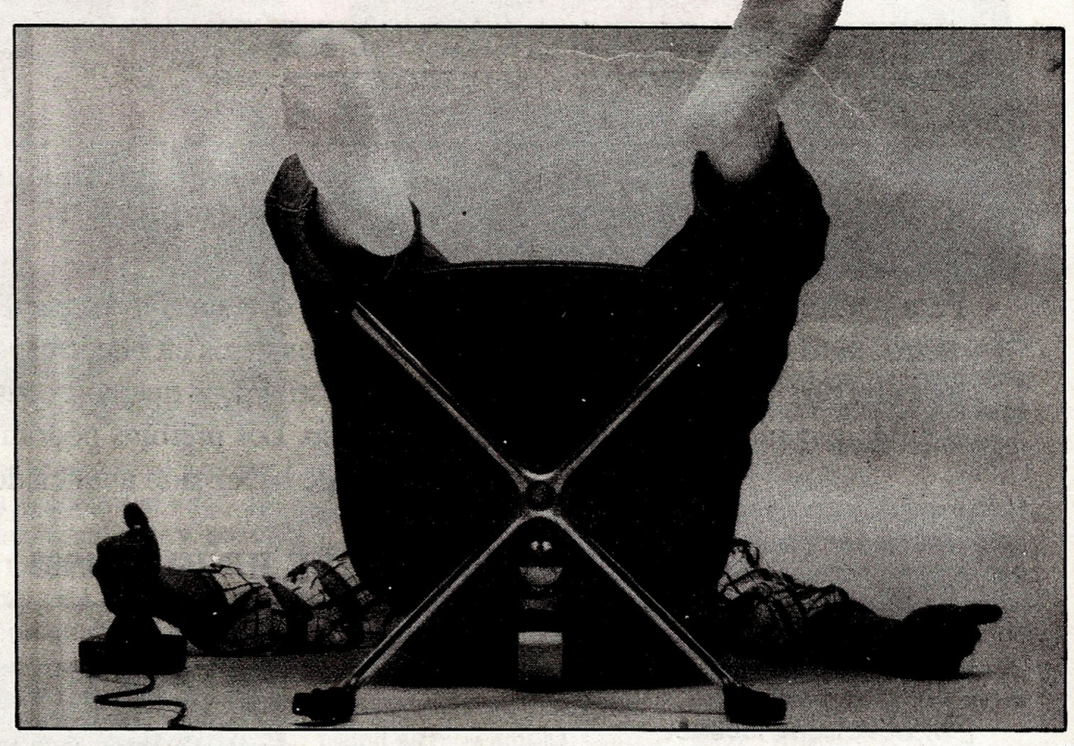
PREMIO



Un soft a eleccion entre IDEA

EL USUARIO JUGUETON

¿Hasta dónde es beneficioso el contacto continuo con la televisión en la formación de un adolescente? Y aún más, ¿cuál es la incidencia de los video juegos con el clásico esquema del "matamarcianos" en la formación de niños sin iniciativa? Ante un mercado irracional, lo que se pretende es que exista un criterio correcto en la elección del software. En definitiva, darle al "jueguito" el lugar que le corresponde dentro de la informática.



A partir de la tesis de Piaget, se sabe que el juego es una de las pocas maneras por la que los chicos, en sus primeros años, aprenden acerca del mundo que los rodea. Las acciones involucradas en los juegos constituyen vivencias que el individuo asimila, y que luego se transforman en pensamientos y abstracciones. La importancia del estímulo en la niñez, fundamentalmente a través de los juegos, se hace evidente.

Un chico triste, sin sueños ni multiplicidad de experiencias se convierte en un adulto indiferente y mediocre. Creatividad e inteligencia están directamente relacionadas con ese tipo de incentivos a temprana edad. Al resultar el juego una acción agradable y llevadera, el aprendizaje se realiza en forma inconciente. A través de ellos, los chicos conocen el universo, es decir, tanto el conjunto de los objetos (formas, colores, movimientos) como el de los vínculos entre ellos (relaciones y abstracciones). Armar casas con módulos de plástico encastrables, o meterse en la cocina a experimentar mezclas extrañas como el más famoso de los científicos constituyen vivencias estimulantes que forman la mentalidad del ser humano.

Formación y Deformación

Por otra parte, son conocidas las críticas que hacen los investigadores de la mente al aparato de TV. Un chico prendido todo el día a una televisión puede perfeccionar su capacidad de percibir o de recibir (hasta ahí nomás), pero nunca la de crear y emitir; jugar (¡ojo!, depende de qué juego) exige iniciativa, creatividad, decisión y superación. Por el contrario, las televíctimas devienen rápidamente en excelentes estudiantes, ejecutantes, y receptores de ideas.

En las escuelas y universidades se convierten en los "estudiosos" y prejuiciosos que reproducen en los exámenes hasta la entonación de las lecciones del profesor.

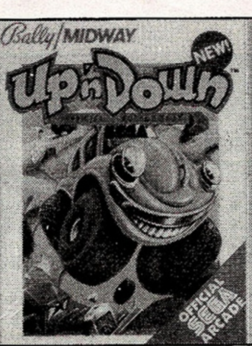
Pueden llegar a tener las mejores calificaciones porque retransmiten lo que recibieron sin integrarlo, pero carecen de iniciativa y jamás cometerían el acto blasfemo de dudar de las cosas que dicen "los que saben", o aventurarse más allá de lo que les dan . No son mediocres de conocimiento pero si de espíritu. Se basan en el principio de autoridad, no son capaces de armonizar sus propios criterios o convencimientos con lo que dicen los demás. Son alumnos "espejo", pueden triunfar como eruditos porque amontonan conocimientos, pero fracasan rotundamente como innovadores.

Si alguien es un receptor, le aconsejamos que frene, e intente pensar un poco a cerca de todo esto. En caso de dudarlo, inspeccione su computadora y los programas que usa. Si más del 60% son juegos, se requiere e un exorcista.

Las estructuras mentales

Cuando Piaget se refiere al juego, no habla de ese redundante residuo radiactivo (R.R.R) que ocupa gran parte de sus medios magnéticos, sino de las características que subyacen en los juegos y que obligan a los chicos a ejercitar determinada zona de su mente.

Los juegos, en general, presentan una dosis de acciones que, para ser realizadas , exigen generar en la mente "formas sin contenido". Es decir, estructuras de









pensamiento no aplicadas a cosas concretas, que se forman en el momento de jugar y son rellenadas con las situaciones del entretenimiento. A medida que los chicos juegan, reafirman la estructura.

Si la estructura perdura en el tiempo, las actividades que requerirá para su comprensión esa estructura mental, encontrarán un basamento cómodo al cual rellenar.

Por lo general cuando la gente juega se divierte; y toda diversión resulta agradable. Mientras una persona pasa un buen rato, crea "terreno de cultivo" para plantar, y más tarde, semillas que se adaptarán especialmente a ese terreno. Si los juegos (no necesariamente juguetes) son muchos, habrá diversos abonos para muchas semillas diferentes.

Pero, evidentemente, así como todos los juegos no tienen la misma carga de diversión, tampoco tienen la misma capacidad de preparar un sustrato fértil y estimular las cabezas para posteriores acciones intelectuales. Nosotros (y/o nuestra familia) podemos estar jugando con esa clase de divertimentos que le quitan energía y no aportan casi nada a nuestra existencia.

La mayoría de estos juegos (R.R.R) se agrupan en conjuntos que evolucionaron de un mismo tronco. Uno de ellos está formado por juegos del tipo de los que matan marcianitos, naves, pelotitas, etcétera. Podemos creer que por tener 5 de esos, ejercitamos diversas funciones del encéfalo; pero no es así. Lo único que cambia en cada uno de los cinco es el color y el ruido. La esencia (concepto sobre la cual están basados es la misma).

Cuando elegimos un juego para la casa (y para los más chicos) intentemos percibir las abstracciones que están involucradas en ellos. Y tratemos de optar por los que más les convengan conforme al objetivo planteado. Abstracciones serían para el caso: simultaneidad, equilibrio, causalidad, necesidad, sucesión, suficiencia, estructura, implicación, modularidad, orden, etcétera.

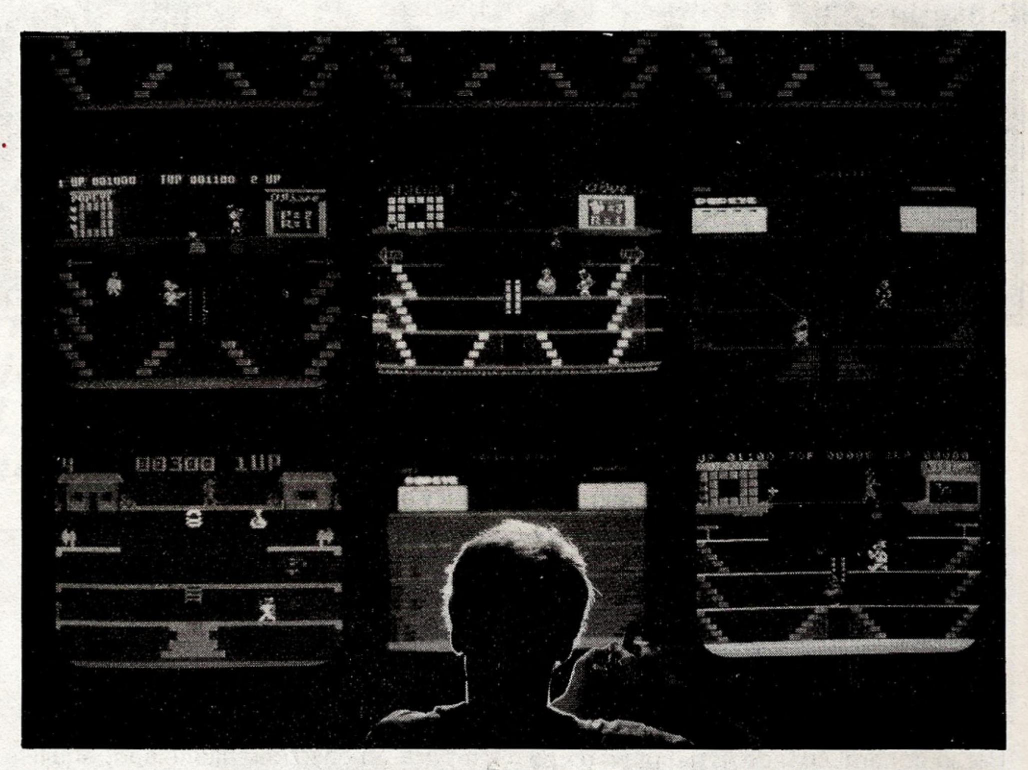


Desde el día en que comencemos a buscar en los juegos esto que comentamos, se abrirá una dimensión nueva en el uso de nuestra máquina. Podremos conciliar, buenos frutos con diversión. Y entonces extender el concepto a las demás actividades de la vida. Pero no abusemos. Todavía hay propensión a razonar de manera obtusa.

Que el juego implique un aprendizaje inconciente y que hayamos conseguido "buenos" juegos para la computadora, no nos autoriza a obligar a nuestros hijos (o a nosotros mismos) a pasárnosla todo el día frente a la máquina.

La computadora tiene la ventaja de poder crear situaciones que en la vida real nunca se darían y así formar estructuras de pensamiento que por otros medios serían muy difíciles de obtener. Pero eso no es lo único que existe.

Otros juegos que no requieren de electricidad son tan instructivos como los computados. Por otra parte, tengamos en claro que no porque nuestro hijo esté jugando 5 horas por día con



juegos que involucren estructura, va a ser un excelente ingeniero.
Hay una época para cada cosa. Hay una edad para cada tipo de juego y un tiempo con determinado juego para un óptimo resultado. Un exceso de ajedrez, por ejemplo, forma una estructura mental que sólo responde al ajedrez.
Llegado a un grado de aprendizaje del juego, el incremento de aprehensión de conceptos abstractos utilizables en otras actividades no es más directamente proporcional al incremento de aprendizaje del juego. Los nuevos conceptos involucrados en el desarrollo de una partida de ajedrez de nivel muy avanzado pasan a servir casi solamente para esa partida. Las estructuras mentales se hacen tan específicas que no se pueden trasladar del juego a otras áreas.
A través de la computadora podemos rellenar "huecos" que antes quedaban al descubierto. Pero esto exige una reestructuración general del método para que la inserción de la computadora en casa como un elemento más de estímulo se haga dentro de un plan coherente. No piense que hay que tener una agenda con horarios para determinadas cosas. Pero antes de hacer algo o de dejar de hacerlo, pensemos si cumplimos con los requerimientos planteados. Un campeonato de PacMan en casa, cada tanto, no es malo; pero programarlo (con todo el peso informático que tiene la palabra) y obligar a la familia a que se prenda a la competencia aludiendo que es bueno para las neuronas no tiene ningún sentido. Piaget no perseguía a sus hijos para que estos jugasen con determinado juego regularmente y adquiriesen así determinadas características.

El mercado mal educado

Por desgracia, nuestro mercado no incita al uso racional del software. La piratería es tal que por pocos australes se puede conseguir cualquiera de las novedades en la materia y es obvio que se las llevan a montones.
Paradójicamente los utilitarios de cualquier clase valen unas diez veces más. Es incomprensible en este tipo de mercado que algo que poca gente compra sea caro.
Si los comerciantes fueran los distribuidores, creadores y productores del programa, y su proyección de venta fuera reducida por la especificidad, con un costo igual al del juego, se justificaría. Pero se dan el lujo de cobrar hasta 30 o 40 australes por copias de programas que consiguen de la misma manera por la que les llegan los juegos que venden a 2.
Entre nosotros, así como la gente en una exposición se lleva folletos de las cosas que no le interesan, sólo por el hecho de que se puede y es gratis, es posible que la gente lleve juegos (de todos, buenos, malos) porque están baratos, nada más. Si los compiladores de Cobol o C estuvieran también a 2 autrales, probablemente mucha gente los compraría, intentando desarrollar pronto tantos programas como ideas de programas. Al no tener tiempo, ganas, ni interés en confeccionar sus propios programas-elefante, correrían al mercado para exigirlos, y las empresas de computación o comerciantes se preocuparían por hacerlos aparecer.
Si los juegos estuvieran más caros, la gente dudaría en comprar tal o cual video game y realizaría una selección más o menos racional. Como mucha gente se cansa rápidamente de los juegos (o aprovechan a cansarse porque están baratos), el consumo de estos se traduce en un negocio infernal, con competencia y baja de precios que a su vez realimentan las ansias de jugar.
El país se convierte así en una verdadera timba y se desalienta la creatividad.

FINAL

La próxima vez que vayamos a comprar un juego, tengamos en cuenta si la disponibilidad de tiempo del usuario permite un engendro más. Si el tiempo que pasamos pegados al joystick es mucho, mejor dejemos la compra para otro momento. Cada tanto revisemos nuestro discos y animémonos a borrar los juegos que ya no nos dejan conformes.
Elijamos los divertimentos para los más chicos pensando en lo que puede llegar a pasar dentro de unos 10 años. La facilidad de aprendizaje de nuestros hijos puede depender entre otras cosas, de eso. No consideremos que la computadora es el único medio para la didáctica lúdica. Los genios de siglos anteriores se formaron sin computadoras, pero muy probablemente hayan tenido infancias estimuladas por juegos y vivencias en general muy "piolas".
Lo que se pretende es que sepamos elegir qué juegos, y la manera de usarlos para que puedan estimular a la gente que nos rodea, en caso de que surja la idea de jugar a algo con la computadora. A través de esa maquinaria, podremos lograr la creación de situaciones que por otros medios serían imposibles de crear. Y a partir de eso dar pie a esquemas que generan diversos aspectos de la formación de una persona, desde la inteligencia (razonamientos abstractos, cálculos numéricos, agilidad mental), hasta la honestidad (no hacer trampa cuando se puede), la humildad (no agrandarse y burlarse de los demás cuando se gana algún juego) la superación (acostumbrarse a elaborar sanamente el hecho de haber perdido un juego para no caer en frustaciones profundas) y otros aspectos más.

Alejandro Parise

4to CONCURSO

DE PROGRAMAS

auspiciado por TELEMATICA S.A. que proveerá los siguientes Premios:

PRIMER PREMIO

UN PERIFERICO

(a elección entre un monitor, y una disquetera y una impresora).

SEGUNDO PREMIO

UN PERIFERICO

(a elección entre un monitor, una disquetera y una impresora).

ESPECIAL

Entre los programas recibidos, algunos de ellos podrán ser editados por SYSTEMAC S.A., reconociéndose los derechos de autor.

Se premiará el mejor software de cualquier clase (juegos, utilitarios, científico o comercial).

BASES

No sólo será indispensable que el programa enviado en casete ó disquete funcione correctamente, sino que además debe cumplir con ciertas reglas:

.Programación estructurada en bloques fácilmente diferenciables.
.Fácil seguimiento del mismo y detalle de éste como parte de su documentación. (Diagrama de bloques con los números de línea que los identifiquen).
.Aclaración y clara explicación de los algoritmos utilizados, deben figurar como parte de la documentación.
.Las variables y/o direcciones de memoria utilizados también se deben incluir en esta documentación.
.Listado de nemónicos assembler y la localización en memoria si es que se utiliza este tipo de lenguaje.
.Calidad y originalidad de gráficos, sonidos y pantallas de menú.

Los trabajos deberán enviarse antes del 30 de julio próximo (cierre del certámen) a: Paraná 720, piso 5, (1017) Capital Federal.

INTERPRETES Y COMPILADORES, ¿SON ENEMIGOS ?

¿En qué casos se justifica el uso del compilador ? ¿Cómo se usa ? ¿Qué diferencia hay con el intérprete?
Aquí contestamos estos interrogantes tratando de poner luz sobre este tema.

Hoy en día se encuentra bastante difundida la forma de trabajo interna de una computadora mediante unos y ceros. En cuanto a la visión desde el usuario, es ineludible la utilización de un lenguaje de alto nivel indiscutiblemente siendo el BASIC en el rubro de las microcomputadoras el más difundido. Apuntalan este hecho dos poderosas razones que son, por una parte, su facilidad de uso inclusive para principiantes y, por otra parte, el precio de compra que incluye un intérprete BASIC residente en ROM (memoria de solo lectura) dentro del computador. Esto puede parecer obvio para usuarios de la franja de los microcomputadores, pero no es tan habitual en las clases superiores. En efecto, un computador profesional puede trabajar con multitud de lenguajes según la aplicación principal a que esté destinado. Deberá elegirse el más adecuado para cada tarea con el fin de optimizar la facilidad de su programación, ejecución y tiempo de proceso.

INTERPRETES

Los interpretes BASIC son, en su mayoria, derivados de uno considerado pionero, el CP/M. El resto de ellos conserva curiosamente muchas de las características del mencionado. En el caso del sistema operativo CP/M (Control Programm for Microprocessors) para MSX, el intérprete BASIC no se encuentra incorporado en el computador, sino que debe colocarse el disco que contiene el sistema operativo CP/M antes de prender el computador. Así tendrá lugar el proceso de carga del mismo durante la inicialización de las variables del microprocesador posteriores a cada encendido.
Este procedimiento es semejante al que se debe seguir para la operación con MSX/DOS (MSX Disk Operating System de Microsoft); no debiendo confundirse ambos ya que sus entornos de operación son en cierta medida semejantes pero no completamente iguales. A continuación se tipeará MBASIC seguido de <RETURN> con lo cual se cargará dicho lenguaje en memoria, y se autoejecutará.
Esta simple secuencia de carga en realidad exige por parte del intérprete una revisión de la cantidad de RAM (memoria de lectura/escritura) disponible. Esto se efectúa mediante un sencillo procedimiento: se escribe un carácter clave en toda la RAM disponible teóricamente, para leerla a continuación. El primer carácter leído que no se corresponde con el clave anteriormente escrito implica el fin de la RAM disponible.
Esta cifra es la que aparece incorporada generalmente en alguna de las cabeceras del sistema operativo o del lenguaje cargado.

EDITOR

Un componente muy importante de todo intérprete es justamente su editor, que es un símil de procesador de texto que permite la escritura y corrección del programa disponible en RAM; este proviene de la unidad de casete, disquete o ha sido recién tipeado. Es de destacar que, salvo algunos lenguajes estructurados de reciente generación, cada línea de programa debe llevar un número antecediéndola. Caso contrario el intérprete intentará ejecutarla como un comando inmediatamente, como es el caso de LOAD y SAVE para carga y grabación.
En general, los números de línea no se almacenan en RAM cifra por cifra, sino que se los convierte en un número binario de dos bytes. Esto explica por otra parte por qué no se pueden generar líneas con numeración mayor a 65535 (esta cifra es la máxima alcanzable mediante notación de dos bytes).

TOKEN

También tiene lugar otra "compactación" del texto del programa y es la referida a las palabras claves propias del lenguaje; estas, una vez identificadas, son reemplazadas en RAM por un equivalente numérico particular de cada una llamado TOKEN.

Solo de esta manera es posible el almacenamiento de largos programas ya que el ahorro de memoria conseguido se visualiza fácilmente con solo comparar la longitud de cualquier instrucción BASIC frente a su equivalente de un byte en RAM.
Además solo este método permite efectuar una revisión de la sintaxis de cada línea antes de que sea aceptada por el intérprete como parte del programa en el caso de ciertos "home computers". Durante la ejecución de un programa, el intérprete se comporta como un administrador de direcciones que siempre apunta hacia la dirección de RAM en que se encuentra la próxima instrucción a ejecu-

tarse; cuando encuentra un código token calcula el salto hacia la subrutina de ROM correspondiente para así poder ejecutarla.

SALTOS Y BUCLES

El panorama del intérprete se complica cuando se incluyen instrucciones como GOTO o GOSUB dentro del programa dado que se éste entonces se ve obligado a quebrar su camino secuencial de ejecución, línea tras línea, por un salto hacia un número desconocido en cuanto a su ubicación física dentro de la RAM.
La única solución posible, y de hecho es la real, consiste en recorrer desde el principio del programa las líneas en busca de la indicada. Como es fácil deducir, un abuso en la cantidad de saltos y subprogramas implicará mayor tiempo de ejecución como consecuencia de que a cada bifurcación el intérprete deberá buscar secuencialmente desde el principio del programa la dirección de destino para continuar su ejecución.
El mismo razonamiento vale también para bucles iterativos como es el caso del FOR...NEXT.

LAS VARIABLES Y SUS NOMBRES

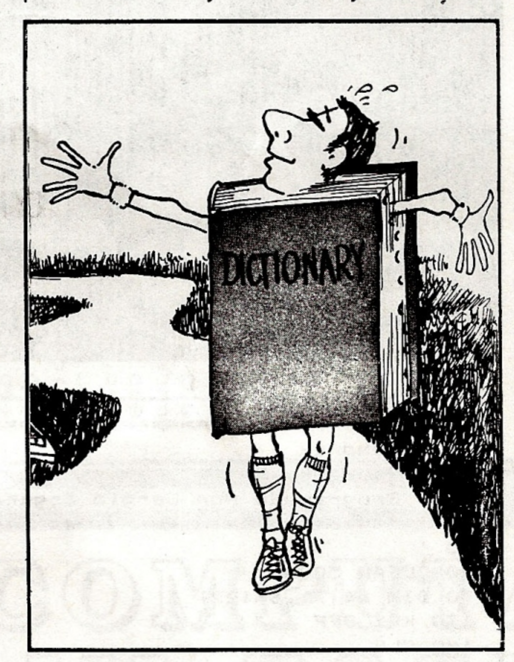
Un método semejante es el empleado para la ubicación de los nombres de las variables usadas en un programa dentro del área de RAM destinada para tal fin. Para cada mención de una de ellas, el intérprete recorre todo el sector hasta encontrarla, caso contrario un indicador de fin de área le hará elevar un mensaje semejante a "Variable no encontrada".
Obviamente este camino tampoco es óptimo en cuanto a su rapidez pero sí en cuanto a su eficacia.

COMPILADOR

La solución a muchos de los problemas arriba mencionados figura como título de este apartado. En efecto, la utilización de un compilador es muchas veces engorrosa, pero el programa resultante se ejecuta sensiblemente más rápido.
Sólo se obtendrá una velocidad mayor si se programa directamente en lenguaje ensamblador.
Volviendo a uno de los temas mencionados anteriormente, no es habitual que en el precio de compra de un computador esté incluido un compilador o, al menos, un buen representante de esta raza de utilitarios. Paquetes originales de algunos de ellos suelen cotizarse por encima del valor de la propia computadora, siendo lógicamente sus mejores exponentes.

EDITOR DE PROGRAMAS

Para escribir un programa en aplicaciones semiprofesionales o profesionales es aconsejable la utilización de un editor distinto del incorporado en el lenguaje usado ya que permite mayores posibilidades frente a los editores de línea; estos son engorrosos en su uso y prácticamente inutilizables en forma rápida y sencilla. WORDSTAR aparece como un editor/procesador muy difundido y renombrado, disponible además para computadores de la norma MSX y computadores de franjas superiores como son los Professional Computers (PC).



En realidad, cualquiera de ellos permite escribir el programa en el lenguaje que entiende el compilador a usar, mientras su almacenamiento sea en forma de código ASCII.

PRUEBA Y ERROR

En el mejor de los casos, un programa escrito sin errores sintácticos será compilado en algunos minutos, pero también cabe la desgraciada posibilidad de un cese de tareas por parte del compilador, el cual devuelve el control al sistema operativo previa emisión de un mensaje de error acorde con la falla detectada. En este último caso no queda otra posibilidad: debe cargarse nuevamente el editor, a continuacion el programa, y proceder pacientemente a buscar el origen del mensaje de advertencia. Es de destacar la poca información que en algunos casos brindan los mismos.
Resulta casi obvio indicar la necesidad de una nueva compilación con la esperanza de no ver aparecer ninguna protesta por parte de nuestro simpático utilitario; esto indicaría (¡por fin!) una exitosa traducción a código de máquina de nuestro programa, pero no implica, desgraciadamente, que no posea errores de lógica.
Hasta este punto obtuvimos un montón de unos y ceros en lugar de nuestro programa; falta un último paso para convertirlo en ejecutable: las instrucciones de bifurcación (GOTO,GOSUB,etcétera) y las llamadas a subrutinas de la ROM todavía no han sido dotadas de las direcciones de ejecución correspondientes, sino que están indicadas en forma relativa. Dicho en otras palabras, saben a que "barrio" de la RAM tienen que ir, no saben la "calle" y "números" exactos.

LINKER/LOCATOR

Junto al compilador se obtienen además otros componentes necesarios para desarrollar las tareas últimamente mencionadas. Precisamente el LINKER es el encargado de revisar el programa compilado buscando llamadas a subrutinas del sistema operativo o bifurcaciones internas y reemplazándolas por un número absoluto que apunta al comienzo de cada sector correspondiente en RAM. Este sector es agregado por el linker al final del programa compilado y extraído de otro componente del paquete compilador: la/s librería/s de subrutina/s (LIBrary).
El LOCATOR es en realidad parte componente del linker. Su misión es definir la dirección de comienzo del código ensamblado ya que de ahora en más siempre deberá ubicarse en el mismo sector de RAM una vez cargado en memoria.

GRANDES VENTAJAS

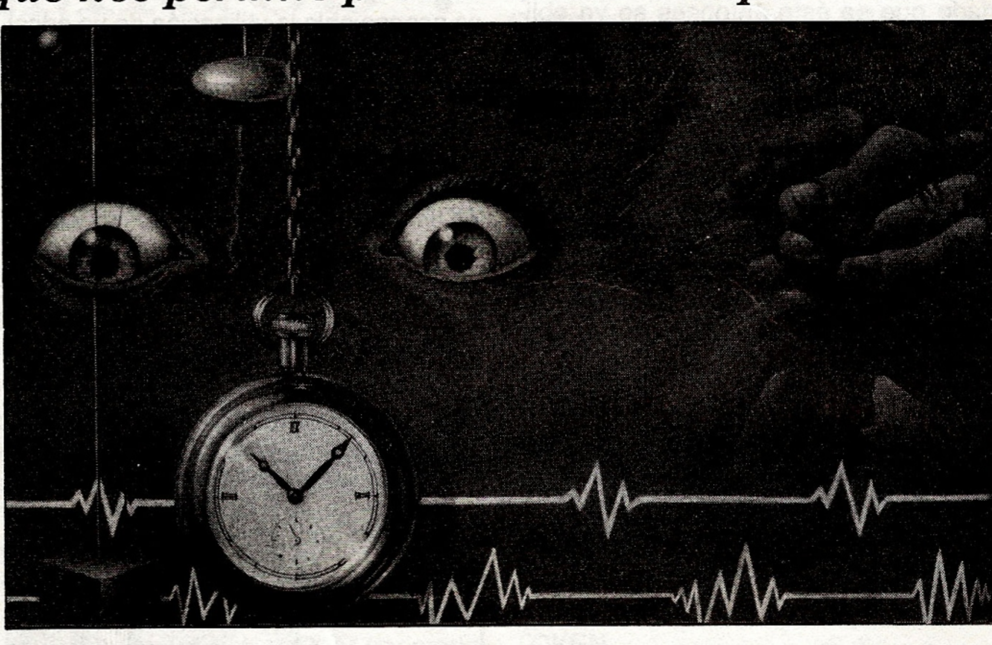
Todo este largo y tedioso procedimiento queda plenamente justificado por una notable aceleración en la ejecución. Esto se debe a que no existen idas y venidas en RAM buscando variables o direcciones sin ubicación conocida.
También se logra una cierta protección frente a otros usuarios que quieran conocer o modificar nuestro esfuerzo de programación siempre que solo cuenten con versiones compiladas y no con los programas fuente. Esto último depende naturalmente sólo de nosotros.

Juan Pablo Bauer

MEMOREX

Clase: Juego
Autor: Sergio Segura

Un programa sencillo que nos permite probar nuestra capacidad de memorizar una cantidad de caracteres al azar.



El mismo consta de 4 niveles (de Novato a Excelente); cuanto más alto sea el nivel , menor tiempo nos dará para memorizar la secuencia de letras. El test se compone de diez ejercicios que, una vez finalizados, serán evaluados de acuerdo con el nivel elegido y cantidad de errores cometidos, brindándonos por último una conclusión al respecto.
Constituye un buen ejercicio para aquellos que tienen dificultad para recordar, cuya memoria es floja y tienen ganas de mejorarla a través de un entretenimiento corto y divertido.
¡Ah!... No se olviden de colocar las teclas en "Mayúsculas" antes de comenzar y ¡Buena Suerte!.

ESTRUCTURA:

10-190: Inicialización de variables.
200-290: Control del Programa.
480-560: Rutina de selección al azar de caracteres.
600-1300: Selección del Nivel de Juego.
1500-1590: Ingreso de las letras y comprobación de errores.
1600-1990: Evaluación y conclusión.
2000-3500: Impresión de la pantalla de presentación.

VARIABLES IMPORTANTES:

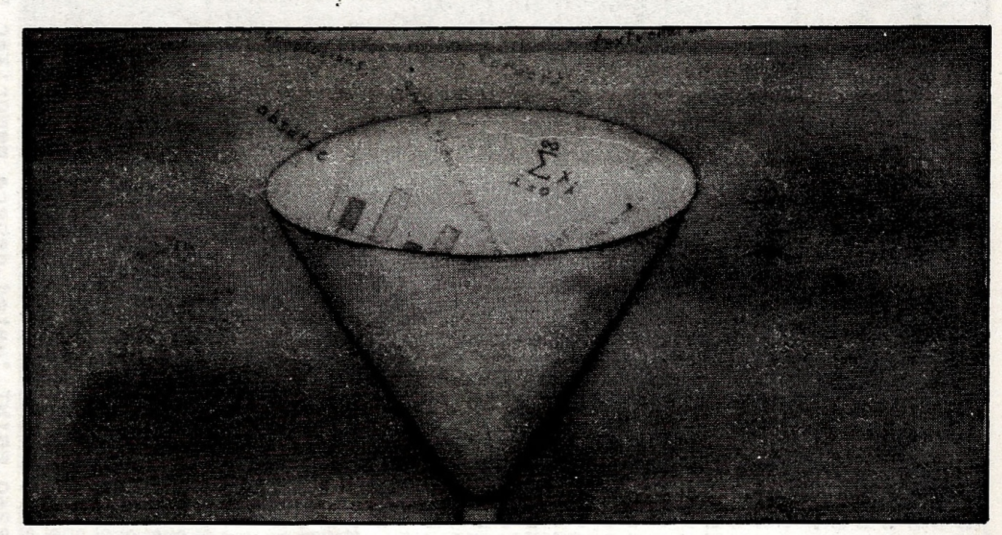
A$(): Caracteres seleccionados.
B$(): Vector que contiene todos los caracteres posibles.
P: Contiene el nivel de dificultad.
T$: Clave a memorizar.
WW: Cantidad de errores.
RR: Cantidad de aciertos.
Aux: Ciclo de cantidad de problemas.
G: Subíndice al azar de B$().

```
10 '================================
20 ' Memorex
30 ' Para Msx
40 ' Programado Por Sergio Segura
50 '================================
60 '
80 CLEAR 5000
90 DIM A$(7),B$(27) :
130 KEY OFF
140 CLS
150 WIDTH 40
160 FOR AA =1 TO 26
180     READ B$(AA)
190 NEXT AA
200 '========Control de programa======
201 '
210     GOSUB 2000      'Pantalla
220     GOSUB  400      'Prueba
230     GOSUB 1600      'Evaluacion
240     LOCATE 4,18
250     LINE INPUT"Desea otra jugada (S/N)";A$
260     IF LEFT$(A$,1) = "S" THEN RUN ELSE CLS : END
290 '
400 '============Prueba==============
410 '
460 FOR AUX = 1 TO 10
470    CLS
475    LOCATE 0,4
480    PRINT USING "Preparese para leer el problema Nro: ##";AUX
490    A = RND(-TIME)
520    FOR D = 1 TO 7
530       G = INT(RND(A)*100)
535       IF G >26 THEN 530
540       A$(D) = B$(G)
560    NEXT D
565 '
600    T$ = A$(1)+ A$(2) + A$(3)+ A$(4)+ A$(5) + A$(6) + A$(7)
610    LOCATE 15,10 : PRINT T$
620    ON P GOTO 1000,1100,1200,1300
1000   FOR AA = 1 TO 2500: NEXT AA : GOTO 1500
1100   FOR AA = 1 TO 1500: NEXT AA : GOTO 1500
1200   FOR AA = 1 TO 1000: NEXT AA : GOTO 1500
1300   FOR AA = 1 TO 600 : NEXT AA
1400 '
```

```
1500 CLS
1505  LOCATE 5,4
1510  PRINT "Ingrese los 7 caracteres que vio:"
1515  LOCATE 15,10 : LINE INPUT X$
1520  LOCATE 4,14
1530  IF X$ = T$ THEN BEEP : PRINT "Correcto!." : RR = RR+1
1540  IF X$ <> T$ THEN BEEP : BEEP : PRINT "Erroneo, La respuesta era: " : LOCATE 15,16 : PRINT T$ : WW = WW+1
1550  LOCATE 1,20 : PRINT "Presione una tecla: ";
1560  A$ = INPUT$(1)
1580 NEXT AUX
1585 RETURN
1590 '
1600 '===========Evaluacion============
1610 '
1620 CLS : LOCATE 4,4
1640  PRINT USING "Fallo en ## Respuestas y obtuvo ## Correctas";WW;RR
1660 LOCATE 4,8
1800 IF WW > 2 AND P >1 THEN PRINT "Intente un Nivel mas bajo de dificultad"
1820 IF WW < 2 AND P <4 THEN PRINT "Bien, Intente un nivel mas alto"
1840 IF WW <6 AND P>2 THEN PRINT "Lo hizo adecuadamente, espero que mejore"
1860 IF WW =10 THEN PRINT "Humm!!...Dejo la leche en el fuego?. Cerro la llave del Gas?. Pago los impuestos?."
1880 IF WW = 0 THEN PRINT "Muy Bien!. Usted es un Genio!
1890 RETURN
1990 '
2000 '============Pantalla=============
2010 '
2200 LOCATE 0,0,0
2210 PRINT "                ---* Memorex *---
2215 PRINT
2220 PRINT "    El objetivo de este programa es "
2240 PRINT "probar su capacidad de Memoria. La Pan-"
2250 PRINT "talla mostrara siete caracteres que "
2260 PRINT "usted debera memorizar rapidamente."
2270 PRINT
2280 PRINT "    Si usted selecciona un nivel alto "
2290 PRINT "de juego menor sera el tiempo de memo-"
2300 PRINT "rizacion y  mayor el tiempo para con-"
2320 PRINT "testarlas. "
2330 PRINT
2340 PRINT "            1 = Novato
2350 PRINT "            2 = Regular
2360 PRINT "            3 = bueno
2370 PRINT "            4 = Excelente"
3380 PRINT
3400 LINE INPUT "Ingrese su Nivel:  ";A$
3410 P= VAL(A$)
3420 IF P <1 OR P >4 THEN BEEP : GOTO 420
3440 RETURN
3490 '
4000 DATA A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z
```

CONTROL DE INGRESO DE DATOS

Es muy conveniente armarnos una librería de subrutinas modulares, que nos brinden soluciones a problemas que se presentan habitualmente en la confección de nuestros programas: manejo de interrupciones, errores, fechas, conversores de números a letras, etcétera. Por ello, aquí proponemos una alternativa útil, especialmente para bases de datos comerciales.



Cuando en nuestros programas deseamos solicitar al operador el ingreso de algún dato y asegurarnos que este sea correcto (ej. en Sueldos: todos los campos correspondientes a las novedades de liquidación y datos del empleado) recurrimos a los comandos inmediatos que nos brinda el MSX BASIC:

1) INPUT "Mensaje: " ;X%
2) LINE INPUT"Mensaje:";A$
3) B$ = INPUT$(10)
4) N$ = INKEY$

Bien, ellas satisfacen nuestras necesidades simples. Pero cuando requerimos de algo más sofisticado nos encontramos con algunos inconvenientes. No controlan la totalidad del teclado, y solo validan si el tipo de variable ingresada es numérica o no. En el caso de equivocarnos, el BASIC nos regresará un mensaje error, que arruinará la estética de nuestras pantallas.

Un buen ingreso de datos requiere de otro tipo de controles:

1 - Longitud del campo
2 - Tipo de ingreso (numérico, afanumérico o fecha)
3 - Si el campo es numérico, cantidad de decimales.
4 - Si el campo es fecha, validar si la misma es correcta.
5 - Permitir únicamente el ingreso de algunos caracteres.
6 - Posición del cursor en columna (x)
7 - Posición del cursor en fila (y)
8 - Control de todas las teclas de edición (Back space, Delete, Cursores, Home, Insert, Esc, Return) :

La subrutina que describiremos a continuación. Realiza todos estos controles y para ello necesita que le enviemos "Parámetros", o sea qué le digamos que es lo que queremos que valide. Además, se realizó una pequeña demostración de su uso, al principio del programa, donde vemos cómo actúa frente a distintos tipos de ingreso.

```
10 '================================
20 ' Rutina de Ingreso de Datos
30 ' Para Msx
40 ' Programado por Sergio Segura
50 '================================
60 '
100 CLS
110 KEY OFF
120     DATA    6,2,2,"Nombre         :         ",2,3,"Direccion:        ",2,4,"Cod.
Post.:",2,5,"Fecha:",2,6,"Importe    : ",2,10,"CONFIRMA (S/N/F)"
500 '--------------------------------
510 ' Pantalla de datos
520 '--------------------------------
530 '
540 RESTORE 120
550 READ B
560 FOR AA = 1 TO B
570     READ X,Y,SS$
580     LOCATE X,Y : PRINT SS$
590 NEXT AA
610 '
1000 '-------------------------------
1010 '      Ingreso de Datos
1020 '-------------------------------
1030 '
1050 ZZ$ = N1$ : A$ ="" : TI$ = "A" : L = 20: ND% = 0 : X =15: Y=2
1060 GOSUB 30000
1070 N1$ = ZZ$
1080 ZZ$ = D1$ : A$ ="" : TI$ = "A" : L = 20: ND% = 0 : X = 15: Y = 3
1090 GOSUB 30000
1095 D1$ = ZZ$
1100 ZZ$ = CP$ : A$ ="" : TI$ = "N" : L = 4: ND% = 0 : X =15: Y=4
1110 GOSUB 30000
1120 CP$ = ZZ$
1200 ZZ$ = FF$ : A$ ="1234567890" : TI$ = "F" : L = 6: ND% = 0 : X = 15: Y
= 5
1210 GOSUB 30000
1220 FF$ = ZZ$
1250 ZZ$ = IP$ : A$ ="" : TI$ = "N" : L = 8 : ND% = 2 : X = 15: Y = 6
1260 GOSUB 30000
1270 IP$ = ZZ$
1300 ZZ$ = CN$ : A$ ="SNF" : TI$ = "A" : L = 1 : ND% = 0 : X = 20: Y = 10
1310 GOSUB 30000
1320 IF ZZ$ = "S" OR ZZ$ = "F" THEN END
1330 GOTO 1000
1340 '
30000 '**********************************
30010 '*  MANEJO DE CAMPOS EN PANTALLA     *
```

Estructura:

10-1330: Demostración de uso
30000-30200: Validación de datos alfanuméricos.
31000-31140: Validación de datos numéricos.
32000-32180: Validación de fechas.

Parámetros a enviar:

ZZ$: Campo a modificar.
TI$: Tipo de la variable (A,N,F).
L: Longitud del mismo. (número entero < 255)
ND%: Cantidad de decimales.(Número entero <8)
X: Posición del cursor en columna.(entero >=0 y <=39)
Y: Posición del cursor en Fila.(entero >=0 y <=23)
A$: Si deseamos que se ingresen solo algunos caracteres, indicaremos en esta variable cuáles son. Para A$ = "", se permitirán todos (hasta 255 caracteres)

Variables importantes:

A: Controla la cantidad de caracteres impresos
B$: Ingreso desde teclado, que posteriormente será analizado.
AN: Año
ME: Mes
DI: Día
DS(): Vector que contiene la cantidad de días de cada mes
SALI %: Señal que indica si el vector DS() fue dimensionado o no.

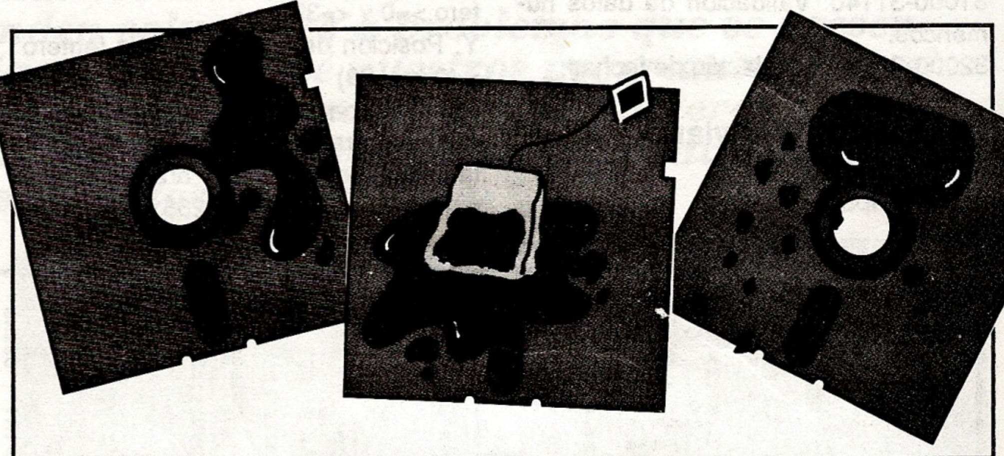
Sergio Segura

```
30020 '*************************************
30030 '
30035 IF ZZ$ = "" THEN ZZ$ = STRING$(L%," ")
30040 IF TI$="N" OR TI$="n" THEN 31000
30050 IF LEN(ZZ$)< L THEN ZZ$ = ZZ$ + SPACE$(L- LEN(ZZ$))
30060 '
30070 '********* alfanumericos ********
30080 '
30090 A=1
30100 LOCATE X,Y,0:PRINT ZZ$:LOCATE X+A-1,Y,1:B$=INPUT$(1)
30110 IF ((A$="" AND ASC(B$) > 31 AND ASC(B$) <> 127) OR (INSTR(1,A$,B$) <>
0)) AND A<=L THEN LOCATE X,Y,0:MID$(ZZ$,A,1)=B$:A=A+1
30120 IF ASC(B$)=127 AND A<L+1 THEN ZZ$=MID$(ZZ$,1,A-1)+MID$(ZZ$,A+1,L-A)+"
"
30130 IF ASC(B$)=8  AND  A>1  THEN  ZZ$=MID$(ZZ$,1,A-2)+MID$(ZZ$,A,L-A+1)+"
":A=A-1
30140 IF ASC(B$)=18 AND A<L THEN ZZ$=MID$(ZZ$,1,A-1)+" "+MID$(ZZ$,A,L-A)
30150 IF ASC(B$)=21  THEN 30080
30155 IF ASC(B$)=27  THEN ZZ$=STRING$(L,32) :A=1
30160 IF ASC(B$)=28 AND A<L+1 THEN A=A+1
30170 IF ASC(B$)=29 AND A>1 THEN A=A-1
30180 IF ASC(B$) <>13 THEN 30100
30190 IF TI$ = "f" OR TI$="F" THEN 32000
30200 Q=INSTR(1,ZZ$," "):IF Q=0 THEN  LOCATE  X,Y,0:PRINT  ZZ$:RETURN  ELSE
MID$(ZZ$,Q,1)="":RETURN
30210 '
31000 '********** NUMERICOS *********
31010 '
31020 ZZ$= STR$(VAL(ZZ$)*10^ND%)
31025 IF LEN(ZZ$)< L THEN ZZ$ = SPACE$(L- LEN(ZZ$)) + ZZ$
31030 IF ND%=0 THEN FO$=STRING$(L,"#") : GOTO 31050
31040 FO$=STRING$(L-ND%,"#")+"."+STRING$(ND%,"#")
31050 A=1
31060     LOCATE    X,Y,0:PRINT       USING       FO$;VAL(ZZ$)/10^ND%:LOCATE
X+L,Y,1:B$=INPUT$(1)
31070 IF ((A$="" AND ASC(B$) >= 48 AND ASC(B$) =< 57 )  OR  (INSTR(1.A$,B$)
<> 0)) AND A<=L THEN LOCATE X,Y,0:MID$(ZZ$,A,1)=B$:A=A+1:
31080 IF ASC(B$)=8 AND A>1 THEN MID$(ZZ$,A-1,1)=" " :A=A-1 :
31090 IF ASC(B$)=127 AND A>1 THEN MID$(ZZ$,A-1,1)=" " :A=A-1 :
31100 IF ASC(B$)=27 THEN ZZ$=STRING$(L," ") : GOTO 31050
31110 IF ASC(B$)=28 AND A>1 THEN MID$(ZZ$,A-1,1)=" " :A=A-1 :
31120 IF ASC(B$)=29 AND A<=L THEN MID$(ZZ$,A,1)="0" :A=A+1 :
31130 IF ASC(B$) <>13 THEN 31060
31135 ZZ$ =STR$(VAL(ZZ$)/10^ND%)
31140   Q=INSTR(1,ZZ$,"  "):IF  Q=0  THEN  LOCATE  X,Y,0:PRINT  USING
FO$;VAL(ZZ$)/100^ND%:RETURN ELSE MID$(ZZ$,Q,1)="":RETURN
31150 '
31180 '
32000 '*********** fechas ************
32010 '
32020   IF SALI% = 1 THEN 32070 ELSE SALI% = 1
32030 DIM DS(12)
32040 DATA 31,28,31,30,31,30,31,31,30,31,30,31
32050 RESTORE 32040
32060 FOR AA=1 TO 12 : READ DS(AA) :NEXT
32070 AN$=MID$(ZZ$,5,2):AN=VAL(AN$)
32080 IF AN<86 THEN BEEP : GOTO 30000
32090 '
32100 ME$=MID$(ZZ$,3,2):ME=VAL(ME$)
32110 IF ME <1 OR ME >12 THEN BEEP :GOTO 30000
32120 '
32130 DI$=MID$(ZZ$,1,2) :DI=VAL(DI$)
32140 IF (AN MOD 4 =0) AND  (ME = 2) THEN DS(2)=29
32150 IF DI<1 OR DI>DS(ME) THEN BEEP : GOTO 30000
32160 ZZ$=DI$+ME$+AN$
32180 RETURN
32190 '
```

(2º parte)

LOS MISTERIOS DEL MSX-DOS - OPERACION

Continuamos develando los secretos de este poderoso sistema operativo. El entorno de trabajo de la disquetera es visto a fondo en esta nota.



Ya hemos analizado qué es un cluster y un sector.

Ahora veremos qué es el DPB, la FAT y el directorio, cómo se utiliza para acceder a un archivo desde el sistema operativo y cómo se usan las famosas "llamadas al sistema" del MSX-DOS.

Utilizando la llamada al sistema 1Bh (GET DPB - Obtener DPB), se pueden obtener estos datos. En el caso de la unidad Talent DPF-550/555 (que es la que se utilizó para realizar esta nota) los valores obtenidos del DPB son:

Número de drive: 0 (el clásico A:)
Identificador de medio: 0FDh
Tamaño del sector: 0200h
Máscara del directorio: 0Fh
Desplazamiento del directorio: 04h
Máscara de cluster: 01h
Desplazamiento de cluster: 02h
Sector tope del FAT: 0001h
Cantidad de FATs: 2
Cantidad de entradas de directorio: 70h (112 entradas como máximo)
Sector tope del área de datos: 00C0h
Cantidad de clusters + 1: 163h
Cantidad de sectores requeridos por FAT: 2
Sector tope del área de directorio: 0005h
Dirección de memoria del FAT: 0E796h

* FAT (File allocation table - Tabla de ubicación de archivos)

La unidad de escritura en el MSX-DOS es el cluster. Los archivos que ocupen más de un cluster (1 kbyte en el DPF-555) se deben escribir en varios clusters. Pero en este caso no siempre se utilizan clusters consecutivos. Especialmente cuando se han creado y borrado archivos muchas veces, los clusters que ya no se utilizan más se disponen más o menos al

Figura 1. Ejemplo de FAT

```
Direcci"n
de comienzo |4 bits|4 bits|
del FAT-->   *-------+-------*
             |  F        D   |-> FAT ID
             *-------+-------*
+1           |  F        F   |\
             *-------+-------* > Dummy (sin uso)
+2           |  F        F   |/
             *===============*
+3           |  0        3   |
             *===============* Entrada de FAT 2:sigue=003h-*
+4           |  4    #   0   |                             |
             *-------+=======*   v-------------------------+
+5           |  0        0   | Entrada de FAT 3:sigue=004h-*
             *===============*                             |
+6           |  F        F   |   v-------------------------+
             *=======+=======* Entrada de FAT 4:sigue=FFFh (fin)
+7           |  6    #   F   |
             *-------+=======*
+8           |  0        0   | Entrada de FAT 5:sigue=006h-*
             *===============*                             |
+9           |  F        F   |   v-------------------------+
             *-------+-------* Entrada de FAT 6:sigue=FFFh (fin)
+10          |       #   F   |
             |        =======*
```

Figura 2

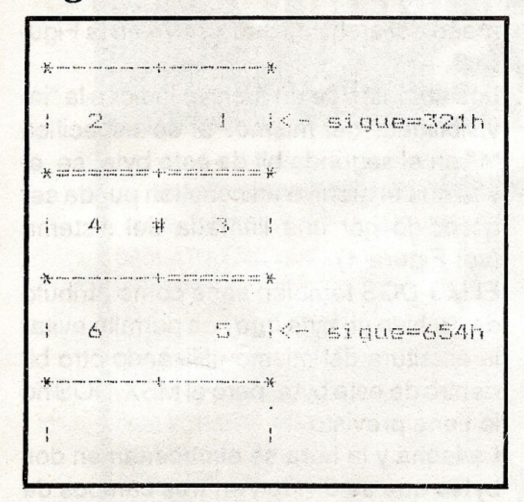


Figura 3- Disposición del directorio

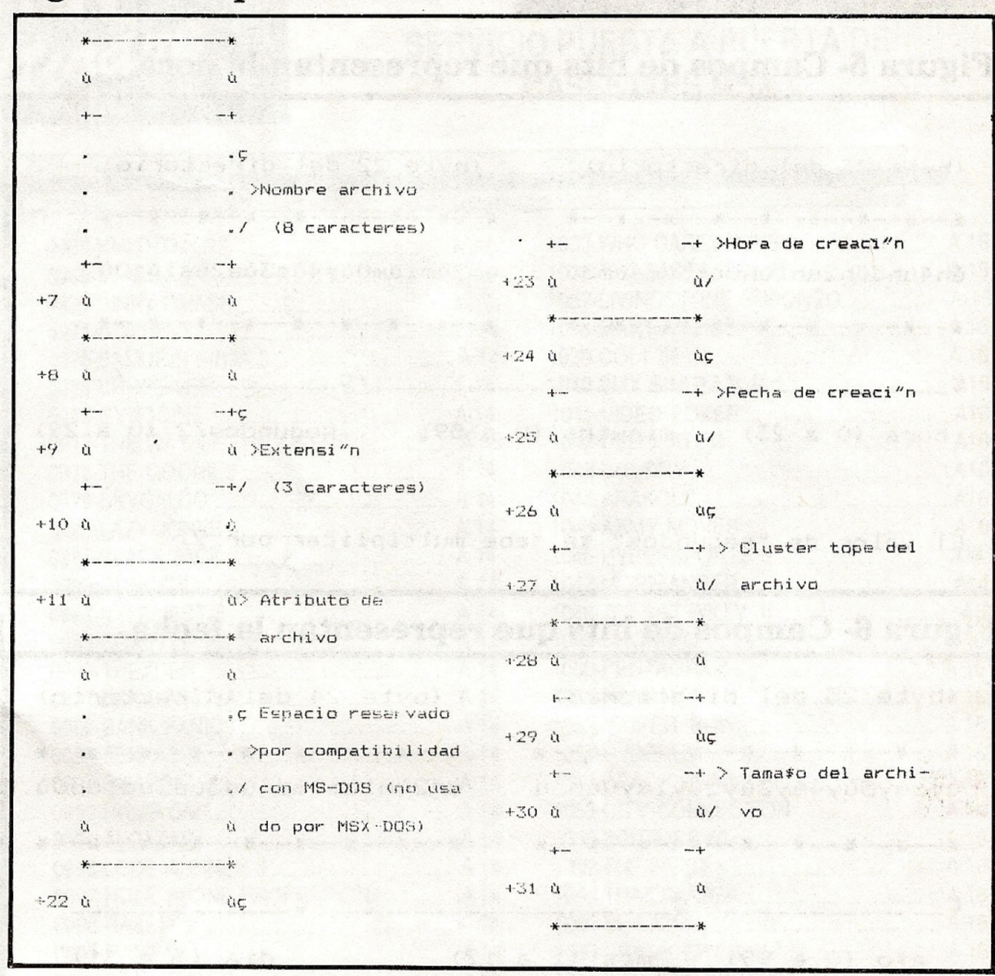
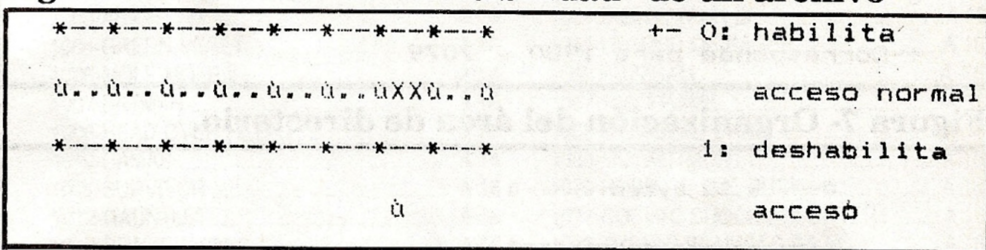


Figura 4 - Atributo de “invisibilidad” de un archivo



azar en el disquete.

Cuando se crea un archivo grande, los datos del mismo se reparten en muchos clusters, y éstos se hallan almacenados donde se encuentre lugar disponible (virtualmente en cualquier posición dentro del disco).

La información que permite reunir todos los clusters que forman un archivo se almacena al principio del disco, a fin de facilitar el acceso a los datos de ese archivo. Esta es la misión del FAT.

Cuando se encuentra un cluster en malas condiciones (no graba bien, por ejemplo) el FAT también almacena esta información para que no se pierdan datos tratando de almacenarlos en dicho cluster.

La información que "engancha" a los datos de un archivo e indica qué clusters no son utilizables es lo que se necesita para poder manejar los archivos de un disco.

Sin esta información el disco entero sería inútil. Por esta razón se almacena más de un FAT dentro del disco para prevenir una pérdida accidental.

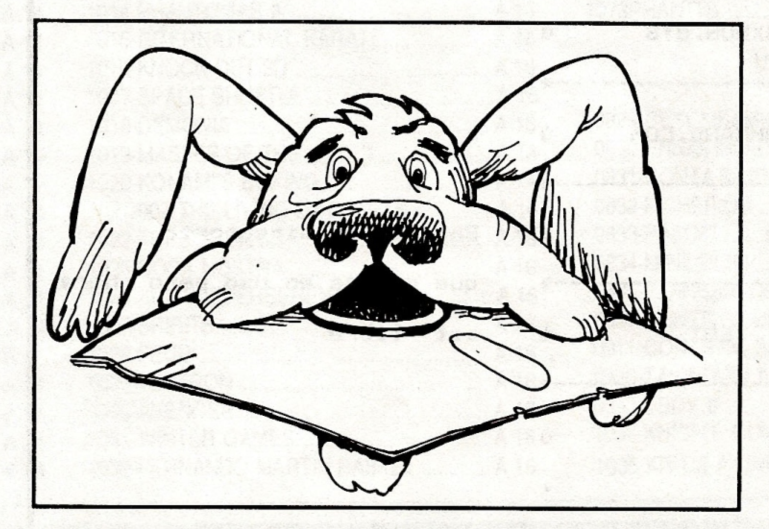
La figura 1 nos muestra un ejemplo de FAT. El primer byte se denomina "FAT ID" y contiene el identificador del tipo de medio utilizado. Los dos valores siguientes no tienen valores significativos.

Desde el cuarto byte (dirección inicial + 3) en adelante, la información de encadenamiento de datos reales se almacena en el formato (algo extraño) de 12 bits por cluster.

Cada área de 12 bits que contiene información de encadenamiento se llama "entrada de FAT". Nótese que la entrada de FAT comienza con el número 2. A cada entrada de FAT le corresponde un cluster. La información almacenada en el FAT debe leerse como se indica en la figura 1.

La información de encadenamiento es el valor que indica el siguiente número de cluster utilizado. El valor FFFh significa que el archivo finaliza en ese cluster. El ejemplo de la figura 1 muestra un archivo con tres clusters (cluster #2 -> cluster #3 -> cluster #4), y el archivo de dos clusters (cluster #5 -> cluster #6).

Hemos utilizado clusters de números pequeños y ordenados para que se entienda mejor. La cruel realidad es siempre más complicada (Figura 2)



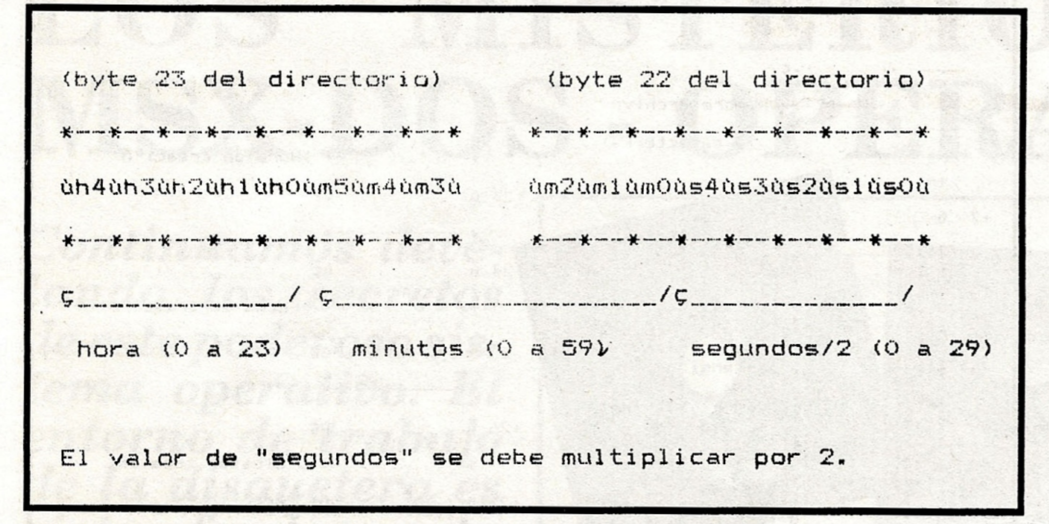
* Directorio

El FAT, como lo describimos anteriormente, contiene la ubicación física de los datos dentro del disco y no incluye información sobre el contenido de los datos escritos allí.

Por lo tanto, se requiere otra fuente de información diferente del FAT para conocer el tipo de datos que se almacenan en un archivo. Esta fuente se denomina "directorio".

Una entrada de directorio está compuesta de 32 bytes y almacena el nombre del archivo, los atributos del mismo, la fecha de creación, la hora de creación, el primer cluster utilizado por el mismo, y el ta-

Figura 5- Campos de bits que representan la hora.



```
(byte 23 del directorio)        (byte 22 del directorio)

*---*---*---*---*---*---*---*   *---*---*---*---*---*---*---*

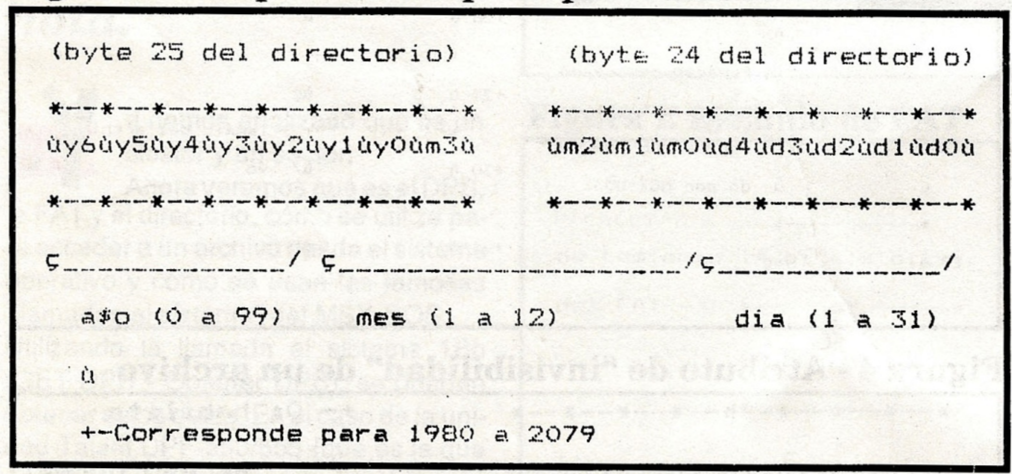
ùh4ùh3ùh2ùh1ùh0ùm5ùm4ùm3ù       ùm2ùm1ùm0ùs4ùs3ùs2ùs1ùs0ù

*---*---*---*---*---*---*---*   *---*---*---*---*---*---*---*

ç______________/ ç____________________/ç______________/
 hora (0 a 23)    minutos (0 a 59)      segundos/2 (0 a 29)


El valor de "segundos" se debe multiplicar por 2.
```

Figura 6- Campos de bits que representan la fecha.



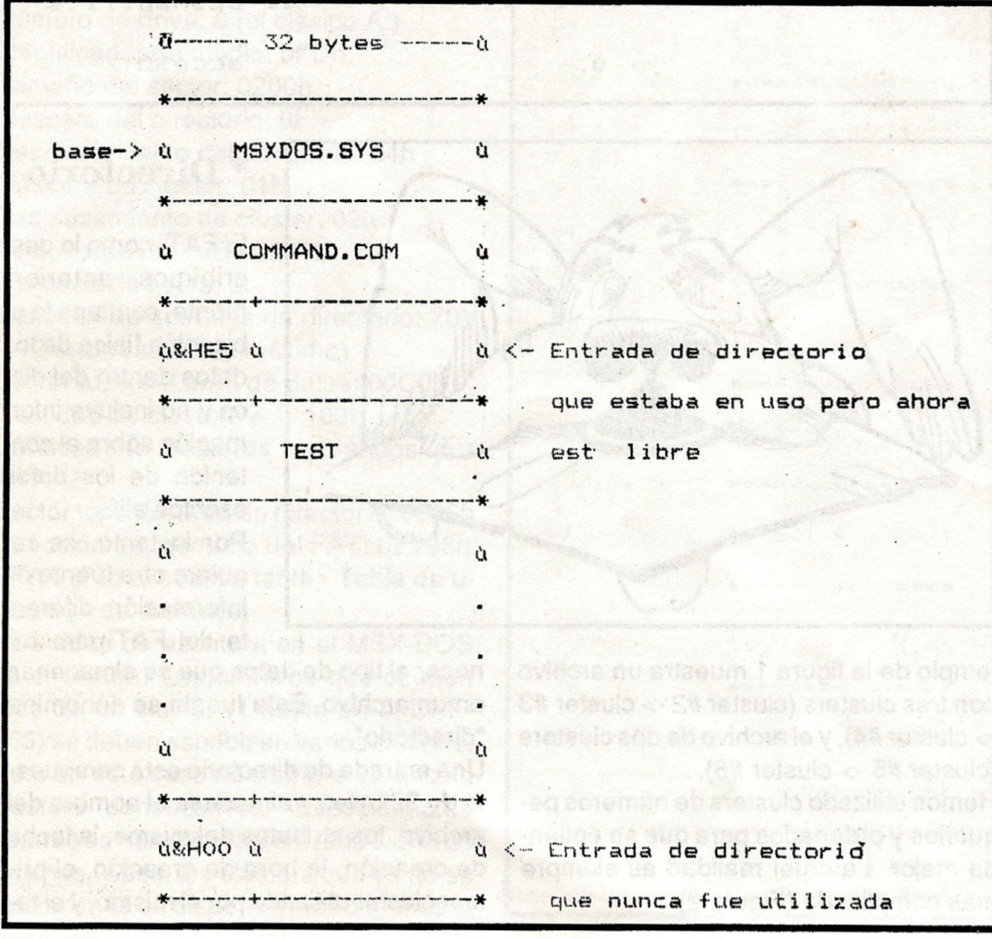
```
(byte 25 del directorio)        (byte 24 del directorio)

*---*---*---*---*---*---*---*   *---*---*---*---*---*---*---*
ùy6ùy5ùy4ùy3ùy2ùy1ùy0ùm3ù       ùm2ùm1ùm0ùd4ùd3ùd2ùd1ùd0ù

*---*---*---*---*---*---*---*   *---*---*---*---*---*---*---*

ç______________/ ç____________________/ç______________/
  a#o (0 a 99)   mes (1 a 12)             dia (1 a 31)
  ù
  +-Corresponde para 1980 a 2079
```

Figura 7- Organización del área de directorio.



```
          ù------ 32 bytes ------ù

          *----------------------*

base-> ù    MSXDOS.SYS           ù

          *----------------------*
          ù    COMMAND.COM       ù

          *----+-----------------*

          ù&HE5 ù                ù <- Entrada de directorio

          *----+-----------------*   que estaba en uso pero ahora

          ù    TEST              ù   est  libre

          *----------------------*
          .
          ù                      ù
          .                      .

          .                      .

          ù                      ù

          *----+-----------------*

          ù&H00 ù                ù <- Entrada de directorid

          *----+-----------------*   que nunca fue utilizada
```

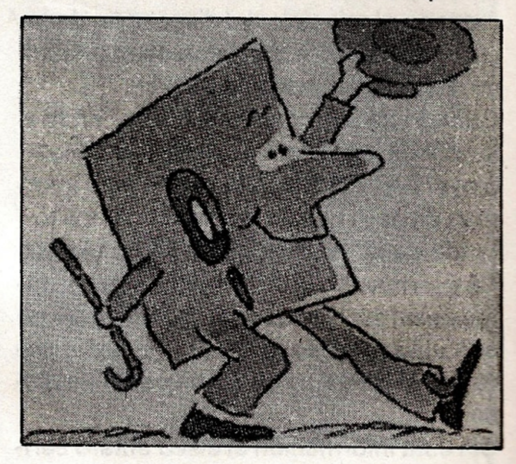
maño del archivo, como se vé en la Figura 3.

Los atributos de un archivo indican la "invisibilidad" del mismo. Si se especifica "1" en el segundo bit de este byte, se evita que el archivo en cuestión pueda ser accedido por una llamada del sistema (ver Figura 4).

El MS-DOS también tiene como atributo de archivo un byte que nos permite evitar la escritura del mismo utilizando otro bit dentro de este byte, pero el MSX-DOS no lo tiene previsto.

La fecha y la hora se almacenan en dos bytes que se dividen en tres campos de bits, como se ve en las figuras 5 y 6. Dado que solo están preparados 5 bits para la hora, la unidad mínima de tiempo almacenable es 2 segundos. El año (1980 a 2079) se especifica usando 0 a 99 en 7 bits.

El lugar donde esta información de directorio se encuentra almacenada es el "área de directorio" en el disco que se



describió en una nota anterior.

La ubicación (sector tope) se almacena en el DPB (Drive Parameter Block - Bloque de Parámetros de la Unidad). Las entradas de directorio (ubicaciones de almacenamiento del directorio) se arreglan cada 32 bytes en el área de directorio, como se indica en la figura 7. Cuando se crea un archivo, el directorio se actualiza utilizando la posición más baja disponible (sin uso) dentro del mismo.

Cuando se borra un archivo, lo que en realidad se hace es grabar un & HE5 en el primer byte de la entrada de directorio correspondiente a dicho archivo, indicando que está vacía.

Cuando se emplean todas las entradas de directorio, no se pueden grabar más datos en el disco, aunque haya mucho espacio libre. La cantidad máxima de archivos que se pueden grabar en una unidad Talent DPF-555 es de 112. Este número se puede averiguar a través del DPB.

Hugo Daniel Caro

MICROBYTE *Software*

MONTEVIDEO 252 - BUENOS AIRES - T.E. 38-0331

SERVICIO PUERTA A PUERTA DE CASSETTES MSX

Título	Precio
0801-ATHLETIC LAND	A 14
0802-HYPER SPORT 1	A 14
0803-HYPER SPORT 2	A 14
0806-ROAD FIGHTER	A 14
0809-FLIGHT PATH 737	A 14
0814- YIE AR KUNG FU-	A 14
0826-KONAMI'S SOCCER	A 14
0827-BOULDER DASH	A 14
0836-STOP THE EXPRESS	A 14
0837-RIVER RAID	A 14
0841-H.E.R.O.	A 14
0847-TENNIS KONAMI	A 14
0850-ANTARTIC ADVENTURE	A 14
0856-F-16	A 14
0874-PINGUILANDIA	A 14
0875-CHOPLIFTER	A 14
0876-GALAGA	A 14
0877-KUNG FU MASTER	A 14
0892-GHOSTBUSTERS	A 14
0898-PAC-MAN	A 14
0904-EGGERLAND MISTERY	A 14
0895-EXERION II	A 14
0908-PIPPOLS	A 14
0920-THE DAM BUSTERS	A 14
0922-GRAND PRIX	A 14
0924-PING PONG	A 14
0928-THUNDER BALL	A 14
0929-COMIC BAKERY	A 14
0931-BOUNDER	A 14
0932-KNIGHT LORE	A 14
0933-NIGHT SHADE	A 14
0935-TIME PILOT	A 14
0937-VALKYR	A 14
0938-ALIEN 8	A 14
0939-GUN FRIGHT	A 14
0941-KING'S VALLEY	A 14
0942-MAGICAL TREE	A 14
0945-FORMATION Z	A 14
0947-MOON PATROL	A 14
0948-ZAXXON II	A 14
0949-ELEVATOR ACTION	A 14
0950-BASEBALL	A 14
0951-YIE AR KUNG FU II	A 14
0953-LAS TRES LUCES DE GLAURUNG	A 14
0954-T.Z.R.	A 14
0955-SCARLET 7	A 14
0960-ZANAC	A 14
0961-EXOIDE Z	A 14
0962-RAID ON BUNGELIN BAY	A 14
0963-SWEET ACORN	A 14
0964-XIXOLOG	A 14
0965-CIRCUS CHARLIE	A 14
0966-HYPER RALLY	A 14
0967-BOSCONIAN	A 14
0968-KNIGHTMARE	A 14
0969-Mr. DO	A 14
0970-PINKY CHASE	A 14
0971-SCION	A 124
0973-SAMURAI NINJA II	A 14
0975-BRUCE LEE	A 14
0976-GYRODINE	A 14
0977-THE WAY OF THE TIGERS	A 38
0978-THE GOONIES	A 14
0979-SKYGALDO	A 14
0980-LAZY JONES	A 14
0981-BLACK JACK	A 14
0983-DUNKSHOT	A 14
0984-B.C.QUEST II	A 14
0985-COASTER RACE	A 14
0986-THEXDER	A 14
0987-EXXA INNOVA	A 14
0989-BANK PANIC	A 14
0990-ZEXXAS II	A 14
0991-GUARDIC	A 14
0992-HANG ON	A 14
0994-ALCAZAR	A 14
0995-LODE RUNNER II	A 14
0997-HOLE IN ONE PROFESSIONAL	A 14
0998-RABBIAN	A 14
0999-RAMBO	A 14
1000-STAR FORCE	A 16
1001-MAGICAL KID WIZZARD	A 14
1003-ARKANOID	A 16
1004-GREEN BERET	A 16
1006-MOLE MOLE II	A 14
1007-BATMAN	A 16
1008-HEAD OVER HEELS	A 16
1010-DAMAS	A 14
1011-SURVIVOR	A 16
1012-GAUNTLET	A 16
1013-DONKEY KONG	A 16
1014-PHANTOMAS II	A 16
1015-INTERNATIONAL KARATE	A 16
1016-KNOCK OUT 3D	A 16
1017-SPACE SHUTTLE	A 16
1018-DEMONIA	A 16
1019-MASTER OF THE LAMPS	A 16
1020-KONAMI'S BOXING	A 14
1021-007 THE LIVING DAYLIGHTS	A 16
1022-TURBO CHESS	A 16
1025-COSA NOSTRA	A 16
1026-AUF WIEDERSEHEN MONTY	A 16
1027-SPIRITS	A 16
1028-FEUD	A 16
1029-VESTRON	A 16
1030-AVENGER	A 16
1031-WINTER GAMES	A 16
1032-FERNANDO MARTIN BASKET	A 16
1033-WHO DARES WINS II	A 16
1036-THE LAST MISSION	A 16
1037-LIVINGSTONE SUPONGO	A 16
1038-MARTIANOIDS	A 16
1039-COLT 36	A 16
1040-BMX SIMULATOR	A16
1041-VIDEO POKER	A16
1042-DESOLATOR	A16
1043-COLONY	A16
1044-KRAKOUT	A16
1045-ARMY MOVES	A16
1046-HYPERSPORTS 3	A14
1047-MOPIRANGER	A14
1048-JET SET WILLY II	A16
1049-CHORO Q	A 16
1050-PENTAGRAM	A 16
1051-NONAMED	A 16
1052- SUPER BOWL	A 16
1053-CYBERUM	A 16
1054-BEACH HEAD	A 16
1055-CITY CONNECTION	A 14
1056-SPIT FIRE 40	A 16
1057-THE HEIST	A 16
1058-TRAILBLAZER	A 16
1059-SEA KING	A 16
1060-SPY vs. SPY II	A 16
1061-SAMURAI NINJA III	A 16
1063-DUSTIN	A 16
1064-DEATH WISH III	A 16
1065-LEGEND OF KAGEE	A 16
1066-FUZZBALL	A 16
1067-10th FRAME	A 16
1068-DYNAMITE DAN	A 16
1069-MUTANTMONTY	A 16
1070-HOWARD THE DUCK	A 16
1071-COSMIC SHOCK ABSORVER	A 16
1072-THINGS BOUNCES BACK	A 16
1073PHANTIS	A 16

UTILITARIOS

Título	Precio
0829-DESENSAMBLADOR	A 42
0830-ENSAMBLADOR	A 42
0831-CONTABILIDAD GENERAL	A 14
0832-FICHEROS	A 14
0833-PASCAL	A 42
0834-MINILOGO	A 22
0835-PLANILLA DE CALCULO	A 14
0887-CONTROL DE STOCK	A 14
0911-CONTROL BANCARIO	A 14
0936-TASWORD II	A 22
0993-EDDY II	A 59
1034-KNIGHT COMMANDER	A 22
1035 KIT DE ALINEACION	A 35

TITULOS: ____________________

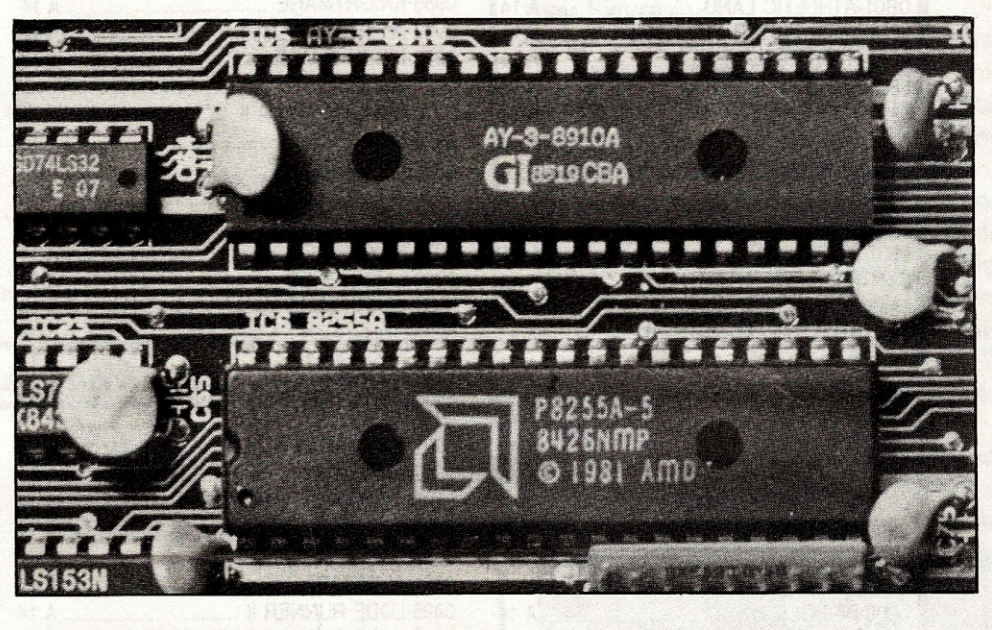
NOMBRE Y APELLIDO: ____________________

DIRECCION: ____________________ C.P. ________ LOCALIDAD: ____________________

FORMA DE PAGO: Cheque/giro   Contrarrembolso   PROVINCIA ____________________

Cheques y giros a la orden de Edmundo A. Goldin. Gastosde Envío A 3 -

HABLEMOS SOBRE LOS CHIPS



En computación, cuando se quiere expresar la idea de "circuito electrónico miniaturizado" simplemente se emplea la palabra "chip" que significa "pastilla", y de esta manera nos vamos entendiendo.

El chip es la denominación que se le da a una pastilla de aproximadamente 40 a 250 milímetros cuadrados y menos de 1 milímetro de espesor que contiene desde unos pocos hasta varios cientos de miles de componentes electrónicos (transistores, resistencias, etcétera).
Los términos "pastilla", "chip" y "circuito integrado" son sinónimos.

EVOLUCION DE LAS PASTILLAS

La industria de semiconductores nació formalmente cuando John Bardeen, Walter Braittain y William Schockley, investigadores de Bell Laboratories, inventaron el transistor en 1947. El transistor fue el primer dispositivo que incorporó materiales semiconductores en su diseño y fabricación.
En la década de 1950, el transistor empezó a desplazar a los tubos de vacío empleados en radio y televisión como amplificadores electrónicos. El transistor también relegó en forma rápida la primera generación de computadoras, las cuales utilizaban miles de tubos de vacío en sus circuitos digitales (el tubo de vacío actuaba como interruptor de conexión/desconexión). Con su ayuda, en la década de 1960 se redujo drásticamente el tamaño de las computadoras.
El siguiente paso fue un concepto aún más evolucionado que el revolucionario concepto de transistor: el circuito integrado. Este es solo un circuito electrónico completo, fabricado en una sola operación, que incluye todos los transistores y componentes requeridos, así como las interconexiones necesarias de acuerdo con un orden preestablecido. Jack Kilby (en Texas Instruments) y Robert Noyce (en Fairchild Semiconductor) concibieron el circuito integrado a fines de la década de 1950.
Desde las primeras pastillas de la década de 1960, el número de componentes

CENTROS DE ASISTENCIA AL USUARIO DE TALENT MSX

CAPITAL FEDERAL

Centro Cultural de la Ciudad de Buenos Aires
Taller Logo de computación: Junín 1930

Fundación de Informática y Educación
Centro de Computación Clínica Asistencia al Usuario Discapacitado: Ramsay 2250 - Pabellón F
Tel. 784-2018

Barrio Norte
Uriburu 1063 - Tel. 83-6892/826-6692

Belgrano
Mendoza 2728 - Tel. 781-2271

Centro
Av. Córdoba 654 - Tel. 392-5328/7611/8043/8051/8251

Flores
Gral. Artigas 354 - Tel. 612-3902

Palermo
Guatemala 4733 - Tel. 71-4124

San Telmo
Chile 1345 - Tel. 37-0051 al 54

GRAN BUENOS AIRES

Castelar
C. Casares 997 - Tel. 629-2247

Lanús
Caaguazú 2186 - Tel. 247-0678

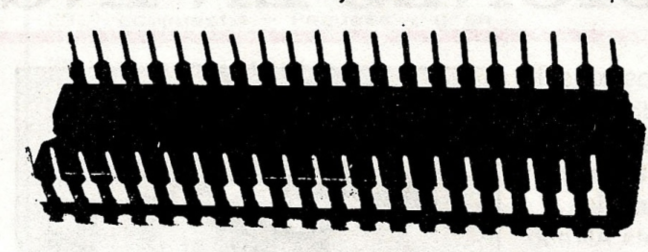
Morón
Belgrano 160 - Tel. 629-3347

Quilmes
Moreno 609 - Tel. 253-6086 al 89

electrónicos que se han miniaturizado a una sola pastilla ha ido aumentando cada año.
En la década de 1980 el espacio físico ocupado por cien mil transistores en una pastilla es el mismo que necesitaba un transistor usado en la década de 1960.
Un subproducto de la miniaturización es la velocidad; al acortarse la distancia que debe recorrer un pulso, se reduce el tiempo de recorrido. Reducir el tamaño de los transistores y las trayectorias hace que el trabajo de conmutación sea más rápido. Las acciones internas de una pastilla se desarrollan en nanosegundos y picosegundos (cienmillonésimas y billonésimas de segundo).
La importancia real de la pastilla radica en sus posibilidades de utilización no solo como hardware de procesamiento (pastillas lógicas), sino también como memoria.
Al ser posible fabricar una computadora completa en una sola pastilla, se abren nuevas aplicaciones de las computadoras en aparatos, juguetes, juegos, todo tipo de dispositivos manuales, etcétera. Además de cierta cantidad de memoria de trabajo, para aceptar los datos del usuario, estas diminutas computadoras contienen memorias permanentes, para almacenar el programa del procesador.
¿Quién hubiera podido pensar en 1951, cuando se presentó al público la Univac I, que treinta años después las computadoras se construirían dentro de una tarjeta de crédito ?

CONSTRUCCION DE UNA PASTILLA

Los circuitos electrónicos en las computadoras digitales son rutas y operadores que conducen pulsos de electricidad de un punto a otro, modificándolos según las necesidades. El flujo de pulsos, a través de los interruptores de conexión/desconexión, se parece a un interruptor eléctrico de pared, excepto porque en el primer caso el interruptor está activado mediante electricidad. El flujo de corriente que sale de un interruptor controla el cierre o la apertura de otros y así sucesivamente. Estos circuitos están diseñados usando lógica booleana. La lógica booleana es un conjunto de reglas que se utilizan para detectar la ausencia o presencia de patrones de pulsos.



Los diseñadores de circuitos usan bloques de construcción de lógica booleana para crear nuevos circuitos digitales. El diseño lógico se convierte más adelante en un diseño electrónico usando componentes tales como transistores, diodos, resistencias, etcétera, que mediante corriente eléctrica ejecutan las operaciones descriptas por la lógica booleana. Existen paquetes de software, para ayudar a efectuar esta conversión. Sin embargo, éste es solo un paso más; el diseño de circuitos integrados determina la forma y colocación de los componentes electrónicos físicos, así como la gran cantidad de interconexiones que servirán como caminos entre ellos. Esta masa de "tuberías" se traza a gran tamaño en los planos, para que puedan ser inspeccionados. Después de que se realizan las correcciones, se imprime una imagen microminiaturizada de la "tubería" en una placa fotográfica y se reproduce cientos de veces para que puedan hacerse simultáneamente muchas pastillas. Las pastillas se construyen en varias capas, empleándose una fotomáscara separada para cada una de ellas.
Literalmente, millones de transistores se construyen en forma simultánea capa por capa.
El material básico para construir las pastillas es el silicio, aunque también se emplean otros materiales como el zafiro y el arsenuro de galio.
El silicio en estado líquido es altamente purificado y después se le agregan impurezas para lograr las características eléctricas deseadas. El resultado se parece a un largo cilindro de alrededor de 10 a 12 cm. de diámetro, que deberá ser cortado en rebanadas delgadas llamadas "obleas".
La fabricación real de la pastilla empieza con la creación química en la oblea de una capa superficial de dióxido de silicio, el cual actúa como aislador. Después se cubre la oblea con una película fotosensible que se expone a la primera fotomáscara. El motivo así transferido se graba en la capa de dióxido de silicio, exponiendo al exterior algunas capas del silicio de base.
A través de varios pasos de fotoenmascarado y grabado, se construyen las diferentes capas de cada componente en la oblea. El paso final consiste en la coloca-

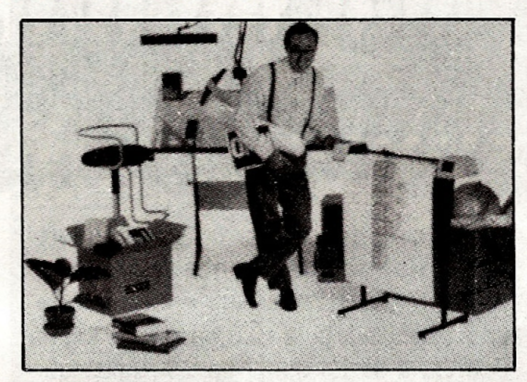
ción de capas de metal en la superficie de la tableta para interconectar entre ellas los transistores y los demás componentes de cada pastilla.
Las pastillas ya terminadas se prueban en la oblea, marcando las defectuosas para eliminarlas. Posteriormente las pastillas se cortan de la oblea y se procede al empaque final, siendo el más conocido la pastilla negra con conectores de "patitas", y entonces se procede a probarlas nuevamente.
La fabricación de circuitos integrados requiere gran precisión. Con frecuencia el número de pastillas que pasan todas las pruebas es menor que las que fallan.

APLICACIONES EN INGENIERIA

A través de la Hot-Line nos consultaron sobre las posibilidades que tiene la Talent MSX para poder realizar programas que tengan aplicación en el área científica y de ingeniería.
El Ingeniero en Construcciones Alfredo Omar López nos ha hecho llegar un ejemplo de aplicación en MSX BASIC, que permite el dimensionamiento a la flexión compuesta de vigas de hormigón armado.

Los valores que se deben ingresar son los siguientes:

Tensión de hormigón (en Tn/cm2)



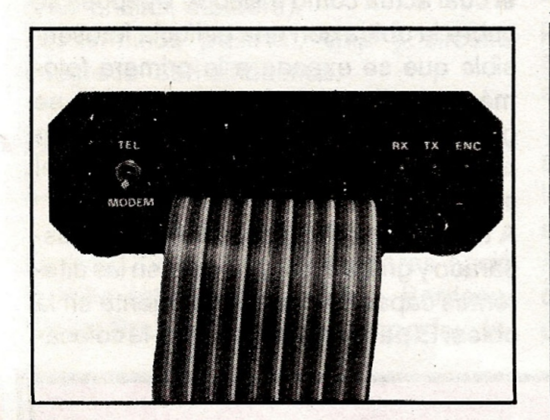
Tensión del acero (en Tn/cm2)
Carga de compresión (en Tn)
Momento flector (en Tn*cm)
Lado mayor columna (en cm)
Lado menor columna (en cm)

Y se obtiene como respuesta:

Sección acero traccionada (cm2)
Sección acero comprimida (cm2)
Tensión arriba (Tn/cm2)
Tensión abajo (Tn/cm2)

Es posible que alguna de las respuestas no se obtengan debido a la configuración particular del problema.
Esperamos que otras aplicaciones como ésta demuestren que el MSX BASIC es una herramienta poderosa para el ingeniero o científico.
Ver Listado 1.

CONTESTANDO A LA HOT-LINE.



Pregunta 1:

¿Cómo puedo reservar memoria en el MSX BASIC para un programa escrito en código de máquina? Sé que el método tradicional es usando la sentencia CLEAR, pero yo quisiera utilizar el área de memoria que comienza en &H8000.

Respuesta:

Supongamos que deseáramos reservar el área de memoria que va desde &H8000 hasta &H81FF. En este caso, el programa BASIC se instalará a partir de la dirección &H8201.
Deberemos ingresar la siguiente instrucción en modo comando (directo):

LISTADO 2

```
10
20 ' Programa para pasar una pantalla
30 ' grabada en LOGO con guardardib
40 ' a una pantalla en BASIC grabada
50 ' con 'BLOAD..,S'
55 '
60 CLEAR 15000:DIM A$(32),B$(32)
70 SCREEN0:WIDTH40
80 INPUT "Nombre del archivo LOGO:";LO$
90 INPUT "Nombre del archivo BASIC:";BS$
100 OPEN LO$ AS#1 LEN=192:FIELD1,192ASA$:SCREEN 2
110 FOR I%=1 TO 32:GET1:A$(I%)=A$:NEXT
120 FOR I%=1 TO 32:GET1:B$(I%)=A$:NEXT
130 CLOSE
140 B1%=BASE(12):B2%=BASE(11)
150      FOR      A%=1      TO      32:FOR      J%=1      TO
192:VPOKEB1%+X%,ASC(MID$(A$(A%),J%,1)):VPOKE
B2%+X%,ASC(MID$(B$(A%),J%,1)):X%=X%+1:NEXT:NEXT
160 LINE(0,0)-(7,191),POINT(0,0),BF
170 BSAVE BS$,0,&H3FFF,S
180 END
```

POKE&HF676,&H01:POKE &HF677,&H82:POKE &H8200,0:NEW
Luego carguemos el programa BASIC y el mismo quedará reubicado como lo deseamos.

Pregunta 2:

¿Cómo puedo hacer para pasar una pantalla grabada con 'guardardib' en el LOGO de MSX al BASIC?
Intenté hacerlo cargando el dibujo con el comando BSAVE "nombre",S (de la unidad de discos) pero no logré recuperarla.

Respuesta:

En efecto, el LOGO guarda en forma diferente las pantallas que el BASIC.
Por eso ofrecemos el listado 2 que permite el traspaso de un formato a otro,siempre que se cuente con una unidad de discos (el comando BSAVE "nombre",S no funciona con casete).

Pregunta 3:

He notado que cargando algunos programas desde casete teniendo mi disquetera conectada, la máquina se bloquea. ¿Es una falla de mi disquetera?

Respuesta:

Lo que sucede con la disquetera es normal. Cuando se inicializa la disquetera, ésta reserva un área de memoria para su propio "solaz y esparcimiento", donde almacena información que necesita para guardar o recuperar archivos.
Si el juego en cuestión excede de una determinada posición de memoria, la rutina de interrupciones de la disquetera (que reside en RAM) queda destruida, y la máquina se bloquea.
Conclusión: apagar la disquetera antes de cargar el juego. Se puede anular la disquetera sin desconectarla pulsando la tecla <SHIFT> cuando se enciende la computadora.

Pregunta 4:

Tengo un modem Talent TMX-510 y un amigo mío también. El modem de mi amigo puede enviar archivos, mientras que el mío no. ¿Cómo lo soluciono?

Respuesta:

En efecto, existe un nuevo "release" (versión) del programa incorporado al modem, que traen los modem más recientes. Esta versión lleva por número el 3.1. Para obtener esta nueva versión, bastará con acercarse a nuestro servicio técnico (modem en mano) para que, sin cargo, se le cambie la ROM con el nuevo "release".

Nuestro horario es: de Lunes a Viernes de 9 a 13 y de 14 a 18:30 hs. en Chile 1347 - Capital

LISTADO 1

```
10 ' Dimensionamiento a la flexion
20 ' compuesta - Peque$a y gran
25 ' excentricidad
30 ' por Ing. Alfredo Omar Lopez
40 CLS
50 INPUT"Tensi"n del hormig"n eB (Tn/cm2):";B:INPUT "Tensi"n del
acero eE (Tn/cm2):";F
60 INPUT"Carga de compresi"n N (Tn):";N:INPUT "Momento flector M
(Tn*cm):";M
70 INPUT"Lado mayor columna D (cm):";D:INPUT "Lado menor columna
B (cm)";I
80 C=M/N/D:IF C<O THEN PRINT"MEJORE eB":GOTO 40
90 IF B<=.07 THEN U=.6:GOTO 130
100 IF B<=.08 THEN U=.7:GOTO 130
110 IF B<=.1 THEN U=.8:GOTO 130
120 IF B<=.11 THEN U=.9
130 X=15/(15+U/B):P=(1.14-X*(1.54-X))/(X*(9.4-4.2*X)-2.52)
140 IF C>=P THEN 160
150 GOTO 240
160 A=-4.2*C-1::V=9.4*C+1.54:W=-2.52*C-1.14
170 X=(-V+SQR(V^2-4*A*W))/2/A:U=15*B/X*(1-X)
180 IF U>E THEN U=E:X=15/(15+U/B)
190 T=((U+15*B)*.95)-U:J=30*(1-X)/X^2:K=6/B/X/(3-X)
200 H=D/1.05:S=N*(M/N+H-D/2):Y=I*H^2/K:IF Y>S THEN 230
210 R=S-I*H^2/K:Z=D-H*.1::G=R/Z/T:F=I*H/J+R/Z/U-N/U
220 PRINT"Secci"n acero traccionada (cm2) FE=";F:PRINT"Seccion
acero comprimida (cm2)-FE=";G;:END
230 F=I*H/J:PRINT"Seccion acero traccionada FE=";F;:END
240
A=-368.367346#*B*I*D/N:V=-17.27891156#*B*I*D/N+12.27891156#+30*C
250 W=-B*I*D/6/N+1/6+C
260 L=(-V-SQR(V^2-4*A*W))/2/A:IF L<4E-03 THEN L=4E-03
270 F=L*I*D:G=1/(1+30*L):O=1/(1/6+L*12.27891156#):P=5*G/3/O
280 S=-N/I/D*(G+C*O):R=-N/I/D*(G-C*O):IF R<O THEN 300
290 IF -S/4<=R THEN PRINT"REDIMENSIONAR":END
300 PRINT"Tensi"n arriba (Tn/cm2) eO=";S,"Tensi"n abajo (Tn/cm2)
eU=";R:PRINT"Seccion acero traccionada FE=";F:END
```

DUMP

Tipo: Utilitario

Autor: Carlos Atashian

Con este programa en lenguaje Assembler se puede inspeccionar por dentro cualquier archivo colocado en disco. Los conocedores del sistema operativo CP/M verán que funciona como un simulador del comando DUMP.

```
;  DUMP40.MAC Simula al comando DUMP del CP/M.
;  SINTAXIS: DUMP40 (Filemame) (START) (P)
.Z80
          ASEG
          ORG       100H
TRAAD1    EQU       0100H
TRAAD0    EQU       0080H
SETDMA    EQU       1AH
BDOS      EQU       0005H
FCB       EQU       005CH
OPENF     EQU       0FH
RNDBRD    EQU       27H
STROUT    EQU       09H
CONOUT    EQU       02H
          LD        HL,ORIGEN
          LD        DE,4000H
          LD        BC,FIN-ORIGEN+1
          LDIR                          ;Reubica el programa a partir de
                                        4000H
          JP        4000H               ;y salida al start
ORIGEN;
 PHASE    4000H
          CALL      VRFDRV              ;Verifica si seespecificó un drive
                                        válido
          LD        DE,TRAAD1
          LD        C,SETDMA
          CALL      BDOS                ;Fija la transfer address en 100H
          LD        A,(FCB+17)          ;Toma segundo parámetro
          CP        'H'                 ;Ve si desea instrucciones
          JP        Z,HELP              ;Si se tipeó 'H' va a imprimir
                                        instrucciones
          CP        'P'                 ;Ve si desea parar en cada página
          JR        NZ,ABRE
          LD        HL,PAGINA
          LD        (HL), 0FFH          ; PAGINA=0FFH, el scroll frena en
                                        c/pantalla
          LD        HL,0000H
          LD        (FCB+33)            ; Fija random record en cero
                                        (comienzo del arch)
          JR        ABRE1
ABRE      CALL      STOHEX              ;Convierte bytes ASCII a nro.
                                        hexadecimal
          LD        (CUENTA), HL        ;y lo guarda en cuenta, que es el
                                        número del
          LD        IX,FCB+33           ;registro que se está leyendo
          LD        (IX),L              ;Este valor lo coloca en el FCB
                                        para indicar
          INC       IX                  ;cuál es el registro inicial
          LD        (IX),H
          LD        A,(FCB+21)
          CP        'P'                 ;Se fija si se eligió la opción "P"
                                        después
          JR        NZ,ABRE1            ;de indicar la dirección inicial de
                                        lectura
          LD        HL,PAGINA
          LD        (HL) ,0FFH
```



Este soft es una versión MSX del programa DUMP para CP/M (que incluye varias mejoras), desarrollada por Carlos Atashian, gerente de Talent en Río Grande.

Una vez ensamblado con el M80 y el L80 dispondremos en el disco de un programa llamado DUMP40.COM. Solo habrá que tipear entonces DUMP40 seguido por el nombre del archivo que deseamos inspeccionar (puede especificarse indicando el drive en que se encuentra).

Inmediatamente comenzamos a utilizar en la pantalla tres campos bien definidos. Hacia la izquierda tenemos, con cuatro caracteres hexadecimales, la posición en que nos encontramos respecto del origen del archivo. En el centro está el contenido del mismo (en hexadecimales) y, hacia la derecha, los caracteres ASCII que lo representan. Para detener momentáneamente el listado puede usarse "control S", y para cancelar "control C".

Otra opción que puede usarse (desde la línea de comando) es la de detener el listado cada vez que se llena la pantalla. Para eso basta con tipear DUMP40 "nombre del archivo" P.

Como opción adicional se puede comenzar el listado desde cualquier posición dentro del archivo. Por ejemplo: si queremos ver el contenido del COMMAND.COM a partir de su byte 256 (0100 en hexadecimal), deberemos tipear DUMP40 COMMAND.COM 0100

Para frenar el scrolling cada vez que se llene la pantalla agreguemos una "P" al final, al lado del último cero (esta vez no hay que dejar un espacio en blanco).

Finalmente, y si no nos acordamos de las instrucciones, tendremos que escribir DUMP40 "nombre del archivo" H, para obtener de esta manera un resumen de las mismas.

```
ABRE 1:   LD      DE,FCB
          LD      C,OPENF
          CALL    BDOS          ;Abre el archivo
          OR      A             ;Acc=O es apertura O.K.  Acc=FFH es error
          LD      DE,ERROR
          JP      NZ,IMPERR
          LD      HL,0000H
;         LD      (FCB+33),HL   ;Random record=0
          LD      (FCB+35),HL   ;Random record=0
          INC     HL            ;(HL)=1
          LD      (006AH),HL    ;Fija largo del registro en 1
          JR      MAS
;
MASREG:   LD      A,(PAGINA)
          INC     A
          JR      NZ,MAS        ;Ve si el scrolling frena cada 23 líneas
          LD      A,(LINEA)
          INC     A             ;Incrementa el contador de lineas
          LD      (LINEA),A
          CP      23            ;Se fija si ya hay 23 líneas en la pantalla
          CALL    Z,PAUSA       ;Va a esperar que se presione una tecla
MAS:      CALL    CRLF          ;Imprime retorno de carro y alim. de línea.
          LD      DE,FCB
          LD      HL,0600H
          LD      C,RNDBRD
          CALL    BDOS          ;Lee 0600H registros del archivo apuntado por DE
                                ;y los transfiere a la transfer address (0100H).
                                ;En HL está la verdadera cantidad de registros
                                ;que se leyeron
          LD      (RESTA1),HL
          LD      (RESTA2),HL
          LD      A,L
          OR      H             ;Se fija si terminó, es decir (HL)=0
          JR      Z,SALIDA
          LD      DE,TRAAD1
;
IMPRAM:   LD      HL,(CUENTA)
          LD      A,H
          CALL    IMPHEX        ;Imprime byte alto  del número de registro
          LD      A,L
          CALL    IMPHEX        ;Imprime byte bajo del número de registro
          LD      A,' '
          CALL    IMPRIM        ;Imprime un espacio en blanco
          LD      BC,0008H      ;BC=8 bytes por renglón
          ADD     HL,BC
          LD      (CUENTA),HL
DUMP8:    PUSH    DE
          LD      B,8
;Impresión de 8 bytes en hexadecimal
DUMP1:    LD      HL,(RESTA1)
          LD      A,(DE)
          INC     DE
          CALL    IMPHEX        ;Imprime byte del archivo
          LD      A,' '         ;Separa los bytes impresos con un espacio
          CALL    IMPRIM
```

```
        DEC     HL
        LD      (RESTA1),HL
        LD      A.L.
        OR      H
        JR      Z,DUMP5
        DJNZ    DUMP1
;Impresión de los mismos 8 bytes pero como caracteres ASCII
DUMP5:  LD      B,8
        POP     DE
DUMP2:  LD      HL,(RESTA2)
        LD      A,(DE)
        INC     DE
        CP      20H
        JR      C,DUMP 3
        CP      7FH
        JR      NZ,DUMP 4
DUMP3:  LD      A,'.'
DUMP4:  CALL    IMPRIM          ;Imprime ASCII de los
                                bytes de archivo
        DEC     HL
        LD      (RESTA2),HL
        LD      A,L
        OR      H
        JR      Z,MASREG
        DJNZ    DUMP2
        CALL    CRLF            ;Imprime ret. de carro y
                                alim. de linea
        LD      A,(PAGINA)
        INC     A
        JR      NZ,IMPRAM
        LD      A,(LINEA)
        INC     A
        LD      (LINEA),A
        CP      23              ;Se fija si imprimió 23
                                renglones
        CALL    Z,PAUSA
        JR      IMPRAM
SALIDA: CALL    CRLF            ;Va a imprimir ret. de
                                carro y aliment. de línea
        LD      DE.TRAAD0
        LD      C,SETDMA
        CALL    BDOS            ;Restaura la transfer
                                address original
        RET                     ;Retorna al MSX-DOS
;Imprime instrucciones
HELP:   LD      DE,AYUDA
        LD      C,STROUT
        CALL    BDOS
        JP      SALIDA
;Subrutina que verifica validez del drive especificado
VRFDRV: LD      A,(FCB)
        INC     A
        RET     NZ
        POP     DE
        LD      DE,ERRDRV       ;Cartel de especifica
                                ción inválida de drive
IMPERR: LD      C,STROUT
        CALL    BDOS
        JP      SALIDA
;Subrutina que imprime en pantalla el carácter leído del archivo
IMPRIM: PUSH    AF
        PUSH    BC
        PUSH    DE
        PUSH    HL
        LD      E,A
        LD      C,CONOUT
        CALL    BDOS
        POP     HL
        POP     DE
        POP     BC
        POP     AF
        RET
;Subrutina que imprime retorno de carro y alimentación de línea
CRLF:   PUSH    DE
        LD      DE,CRLFD
        LD      C,STROUT
        CALL    BDOS
        POP     DE
        RET
;Subrutina que imprime el contenido del ACC en forma hexadecimal
IMPHEX: PUSH    AF
        RR      A
        RR      A
        RR      A
        RR      A
        CALL    PUTHX 1
        POP     AF
PUTHX 1: PUSH   AF
        AND     0FH
        CP      10
        JR      C,PUTHX 2
        ADD     A,'A'-10-'0'
PUTHX 2: ADD    A,'0'
        CALL    IMPRIM
        POP     AF
        RET
;Subrutina que espera que se presione cualquier tecla al finalizar
;una pantalla.
PAUSA:  LD      C,08H
        CALL    BDOS            ;Espera la presión de una
                                tecla cualquiera
        LD      A,0CH
        CALL    IMPRIM          ;Borra la pantalla
        XOR     A
        LD      (LINEA),A       ;Resetea contador de
                                líneas
        RET
;Subrutina que convierte string a dos bytes enteros
STOHEX: LD      HL,0000H
        CALL    STOH1
        RET
STOH1:  LD      B,4
        LD      IX,FCB+17
STOH3:  LD      A,(IX)
        INC     IX
        SUB     '0'
        RET     C
        CP      10
        JR      C,STOH2
        SUB     'A'-'0'
        RET     C
        CP      6
        RET     NC
        ADD     A,10
STOH2:  ADD     HL,HL
        ADD     HL,HL
        ADD     HL,HL
        ADD     HL,HL
        OR      L
        LD      L,A
        DJNZ    STOH3
        RET
;Mensajes
ERRDRV: DEFM 'Drive incorrecto','$'
ERROR:  DEFM 'Archivo inexistente','$'
CRLFD:  DEFB 0DH,0AH,'$'
CUENTA: DEFB 00H,00H
RESTA1: DEFB 00H,00H
RESTA2: DEFB 00H,00H
PAGINA: DEFB 00H
LINEA:  DEFB 00H
AYUDA:  DEFM 'Ejemplo: Para volcar en pantalla el con
             tenido de un archivo',0DH,0
AH
        DEFM 'SAMPLE. TXT a partir de la posición
             0010 (hex) y deteniendo el', 0DH,0AH
        DEFM 'scrolling (en cada pantalla) tipee DUMP40
             SAMPLE.TXT 0010P', 0DH,0AH
        DEFM 'Si desea volcarlo a partir del principio,
             tipee DUMP40 SAMPLE.TXT P',0DH,0AH,'$'
.DEPHASE
FIN:
```

ANIMACION LOGO

Los docentes y usuarios de LOGO en varias oportunidades necesitan incentivar a sus alumnos y niños respectivamente; un método posible es a través de la creación y el juego.

En LOGO es muy fácil crear y también jugar. En primer lugar la tortuguita puede cambiar de aspecto y adquirir el que nosotros deseamos utilizando la imaginación.

Aquí es donde aparece la creación. En cuanto a jugar el primer paso es animar la figura que hemos definido.

En primer término veremos cómo definir la figura. Una vez decidido qué vamos a animar, haremos un esqueleto de dicha figura "observando" cuáles son los movimientos que realiza.

Podríamos tomar como referencia nuestro propio cuerpo cuando caminamos (ver figura 1).

Con la orden edfig podemos volcar nuestro primer esqueleto dentro del LOGO. Cabe aclarar que utilizamos la barra de espacio y los cursores para dibujar.

Una vez definidos todos los movimientos debemos darle movilidad.

La animación la podemos lograr, en forma sencilla, utilizando varias hojas de papel, dibujando en ellas los distintos "esqueletos". Si las hacemos pasar una tras otra frente a nuestros ojos, la ilusión óptica que presenciamos es la de animación.

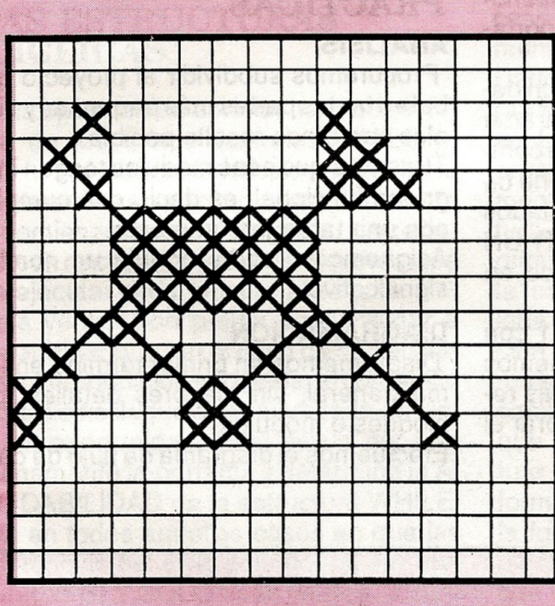
En LOGO la animación la podemos lograr en igual forma. Fijamos la figura en la pantalla al igual que en el papel. Luego, con la orden "esperar" de la fig. 2a, establecemos el período que tardamos en ver la hoja siguiente.

Si queremos que además de moverse en el lugar se desplace por la pantalla, veamos las órdenes de la fig.2b.

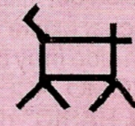
El programa fue realizado por uno de los chicos del Centro Comunitario Nº5 de Isla Maciel, quien observó su perro y lo animó en la pantalla.

Humberto González

FIGURA 1

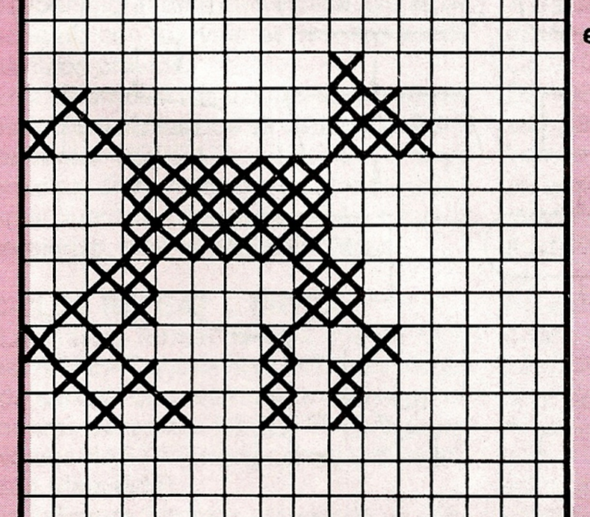


edfig 30

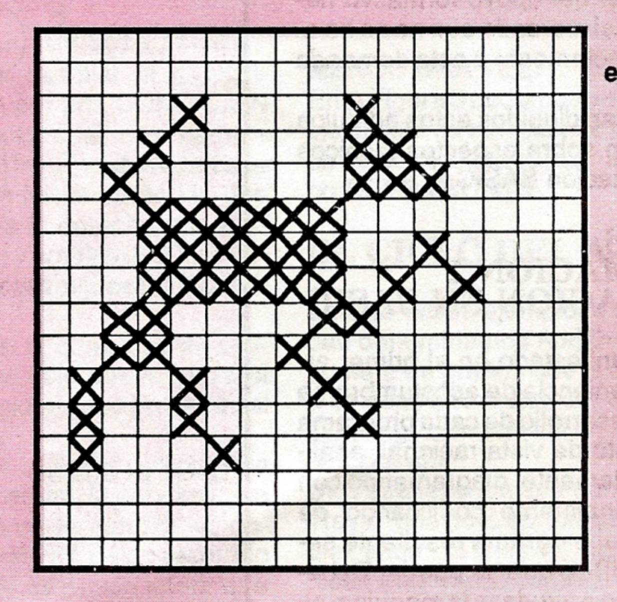








edfig 31



edfig 32

FIGURA 2

2a.

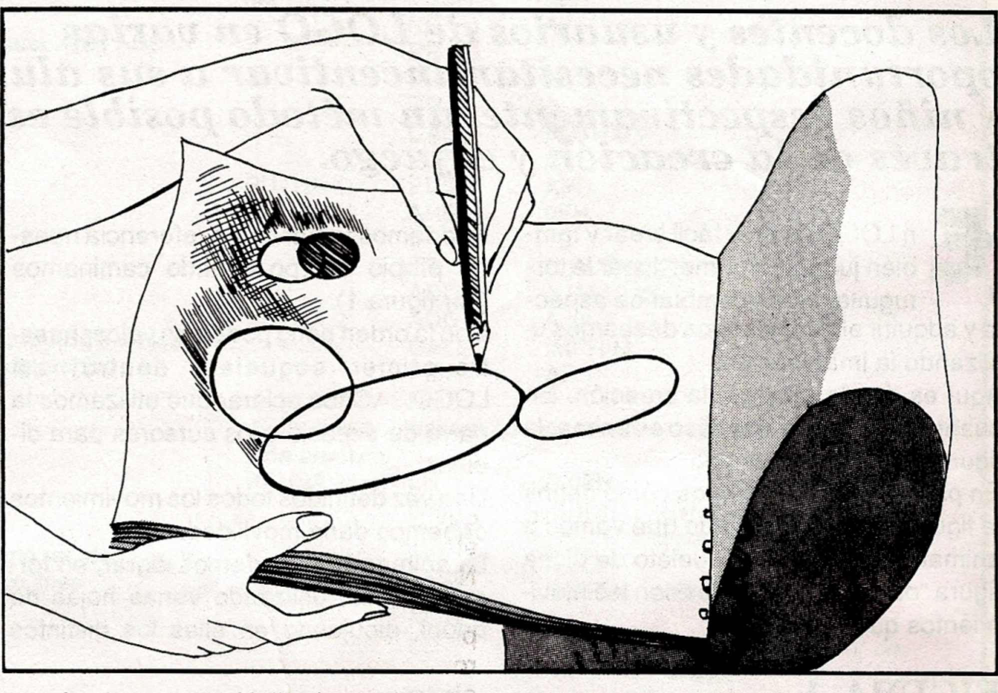
```
para perro
ffig 30
esperar 10
ffig 31
esperar 10
ffig 32
esperar 10
fin
(en el lugar)
```

2b.

```
frumbo 90
sp fvel 12
repetir 40[perro]
(desplazandose)
```

Los ciclos REPEAT ... UNTIL...

En el número anterior publicamos el primer artículo que aborda el tema de las ESTRUCTURAS CICLICAS. El objetivo de esta serie de notas apunta a aportar información y sugerencias a los lectores que puedan inscribirse en el rubro de usuarios serios, en contraposición con los usuarios lúdicos, sin ánimo peyorativo alguno.

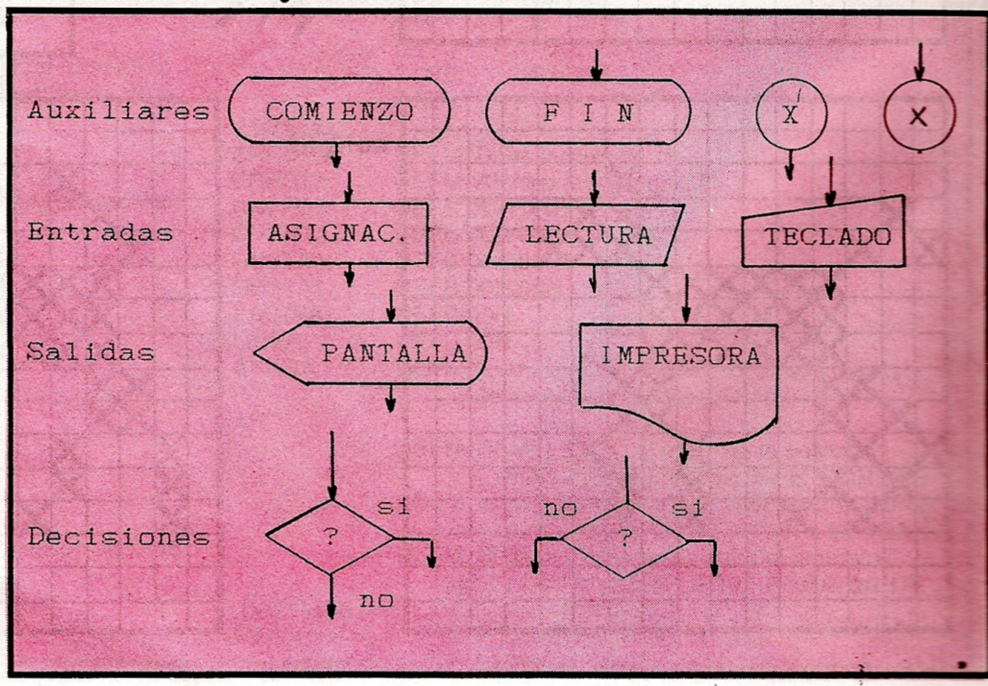


Entre los usuarios serios incluimos a la mayor parte de los adultos, y a una creciente cantidad de adolescentes y jóvenes, que se han acercado a la computadora -sin ser profesionales del área informática- tanto por curiosidad cuasi-científica, como por tratar de incorporar una herramienta más al servicio de la realización más simple, eficiente y rápida de las tareas cotidianas, sean estas de índole laboral o doméstica.
Este nivel de usuarios que desean o necesitan aprender a confeccionar sus propios programas en BASIC, y otros que simplemente quieren deleitarse con la sensación de poderío que brinda la ejecución de tareas con su propio equipo, suelen carecer del apoyo formativo necesario, pues el mercado aun no se ha adecuado correctamente a esta demanda tan especial.
A ellos pues van dirigidos estos artículos de divulgación sobre aspectos técnicos de la programación BASIC.

ANALISIS, DIAGRAMACION Y CODIFICACION EN BASIC

Ya hemos manifestado en el primer artículo la conveniencia de acostumbrarse a encarar el desarrollo de cada programa desde un punto de vista racional, analizando detenidamente, diagramando con prolijidad y finalmente codificando de modo tal que el programa resultante sea lo más LEGIBLE que se pueda. Debemos apuntar no a ayudar a la maquina, sino a hacernos mas fáciles las revisiones, comprobaciones y depuraciones ulteriores, e incluso a facilitar la "transportación" de un usuario a otro.
Si se requiriera "alta velocidad" en un determinado programa, puede prepararse una segunda versión del mismo "de competición", para lo cual existe una serie de recursos, que han sido muy bien tratados en el artículo "COMO ACELERAR UN PROGRAMA EN BASIC" publicado en LOAD MSX de octubre pasado.
Solamente repetiremos la figura 1 con los bloques de diagramacion que hemos de utilizar de manera uniforme, y las recomendaciones básicas para mejorar el "estilo" de programación.

FIGURA 1: BLOQUES



SUGERENCIAS PRACTICAS

ANALISIS

Procuremos subdividir al proyecto global en tantas partes más pequeñas y simples como nos resulte posible.
Tratemos que esos modulos tengan integridad funcional, es decir, que cumplan con una tarea íntegramente.
Asignemos a cada módulo un nombre significativo.

DIAGRAMACION

Diagramemos en primer término en forma general, sin mayores detalles, por bloques o módulos.
Efectuemos el diagrama de flujo de cada

módulo por separado, lo más prolijo y detallado que podamos.
Usemos un bloque para cada sentencia principal que se piense utilizar posteriormente (excluimos los LOCATE, REM, etcétera).
Trabajemos siempre de la misma manera; si copiamos algún programa que no se ajusta a nuestra propia modalidad, adaptémoslo!.
Procuremos el apareamiento BLOQUE (diagrama) - SENTENCIA (programa) en una misma línea.

CODIFICACION

Evitemos, dentro del marco de la lógica, las líneas multisentencias innecesarias.
No usemos "mensajes" dentro del INPUT, usemos un PRINT que lo preceda para ello.
Usemos el LET para las asignaciones y elijamos identificadores fáciles de asociar con el dato.
Usemos REMarks para nombrar cada bloque o módulo.
Indentemos las sentencias interiores de los ciclos y las sentencias alternativas del IF/THEN/ELSE.

LAS ESTRUCTURAS CICLICAS

En la oportunidad anterior tratamos el tema de los Ciclos llamados WHILE ... DO..., cuya característica principal es la de tener la sentencia de control ubicada al comienzo del ciclo, de modo tal que debe ser necesariamente verificada antes de ejecutar cada "vuelta" al circuito.
Esta verificación previa puede llegar a determinar que, en el caso de no satisfacerse la condición impuesta, el contenido del ciclo no llegue a ser ejecutado en ninguna oportunidad. Esto es un aspecto sumamente importante y determina la APLICABILIDAD de la estructura WHILE/DO en todos aquellos casos en que las sentencias del ciclo solo deban ser ejecutadas SI Y SOLO SI se verifica la condición, sobre la base de los datos que puedan ingresarse por teclado, o bien de acuerdo con los valores resultantes de cálculos previos.
En este artículo trataremos el otro tipo de estructura cíclica que se denomina técnicamente: REPEAT ... UNTIL ..., cuya principal diferencia con la anterior radica en el punto de control de salida, no ya al comienzo, sino al final del mismo.

CICLOS REPEAT ... UNTIL ...

La sintaxis formal de este tipo de ciclo es: REPEAT acción/es UNTIL condición (de salida), que podríamos expresar comúnmente como: REPETIR (la/las) acción/es HASTA (que se cumpla la) condición (de salida).
En la Figura 2 hemos preparado un DIAGRAMA DE FLUJO esquemático de un Ciclo REPEAT/UNTIL, en el cual pueden observarse sus características esenciales y donde se han utilizado los bloques convenidos en el artículo anterior.
Nótese como el bloque de control representado por el rombo se encuentra al final del mismo, contrariamente al caso visto anteriormente.
Dado que el BASIC MSX no posee una sentencia que represente explícitamente esta estructura, debemos "exprimirnos la croqueta" para lograr simularla (en computación esta no es una mala palabra). El cambio más notorio que debemos efectuar en la sintaxis formal es modificar la forma en que se plantea la condición que debe ser verificada.
Para diseñar la condición, hemos de cambiar la CONDICION DE SALIDA por una CONDICION DE RECICLAJE, lo cual nos permite cumplir con una de las proposiciones que hemos volcado en la nota anterior, que es la de salir hacia la derecha por el SI (cierto, verdadero) y continuar el flujo secuencialmente hacia abajo por el NO (falso).
Funcionalmente, si la condición de reciclaje se cumple, el flujo deriva hacia el conjunto de intrucciones que preceden al bloque de control.
En el caso de que al llegar por primera vez al bloque de control, la condición de reciclaje impuesta no se cumpla, es obvio que el ciclo en sí ya ha sido recorrido una vez, de lo cual se puede inferir que este tipo de ciclos, se ejecuta siempre AL MENOS UNA VEZ.
En la Figura 3 podemos analizar un ejemplo simple de cálculo que hemos diseñado en base a una estructura cíclica del tipo REPEAT/UNTIL.

FIGURA 2: REPEAT...

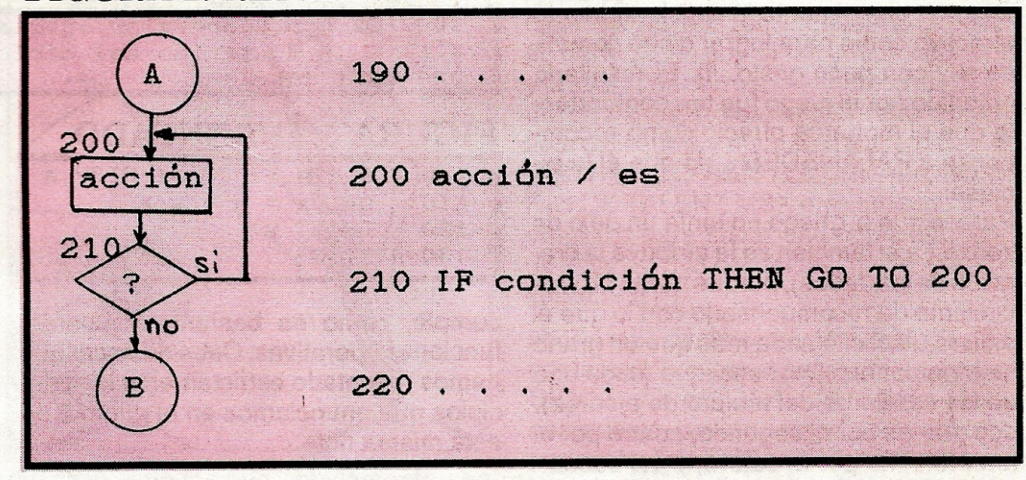
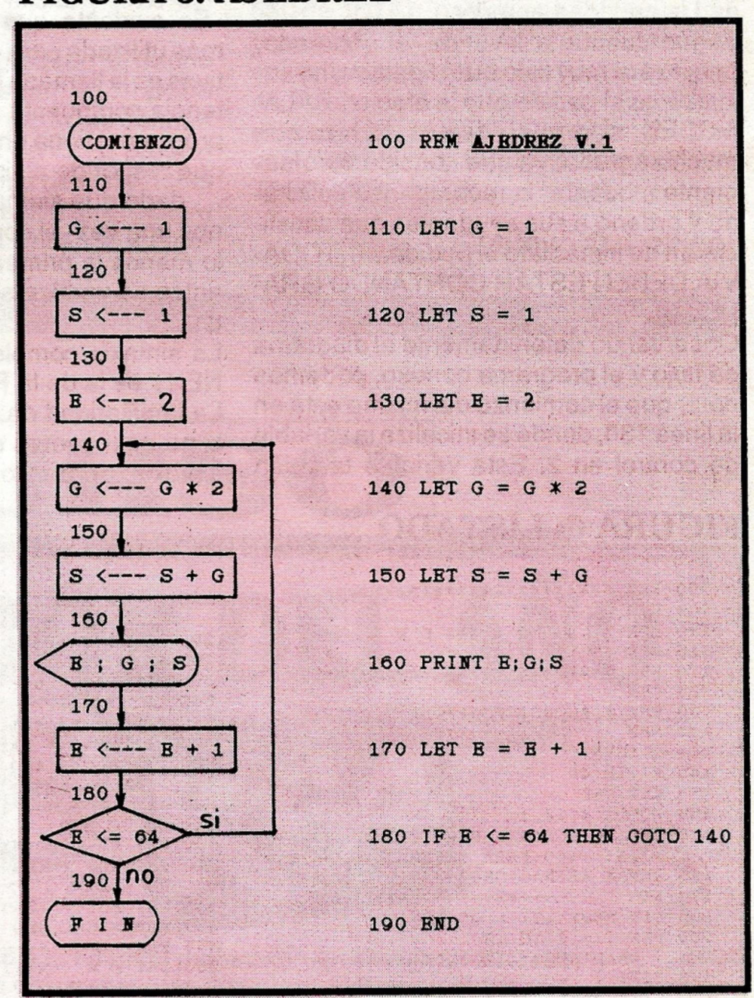


FIGURA 3: AJEDREZ



EL CICLO DEL AJEDREZ

Este conocido ejemplo, que por simplicidad denominamos AJEDREZ, calcula el número de granos que le correspondería al supuesto autor del JUEGO-CIENCIA, según una de las leyendas que rondan por ahí.
Dice esa leyenda que durante el Sitio de Troya - que duró aproximadamente unos diez años - ,a fines de mantener entretenidas a las tropas combatientes entre batalla y batalla, pidióse a PALAMEDES

crear un divertimento lo suficientemente atractivo como para lograr dicho cometido (¡ y con poco gasto...!). El resultado obtenido por el juego fue tan contundente que el monarca ofreció como recompensa a PALAMEDES ...¡lo que él le pidiese!.

Parece que el griego no tenía un pelo de zonzo,(a él también se le atribuye la creación de los dados), porque frente al ofrecimiento de recompensarlo con lo que él pidiera... solicitó nada más que un grano de trigo por el primer escaque (cada uno de los casilleros del tablero de ajedrez), dos granos por el segundo, cuatro por el tercero,ocho por el cuarto, y así sucesivamente, duplicando en cada oportunidad la cantidad anterior.

Según cuenta la leyenda, el soberano, que no era muy listo que digamos, no solo acepto el pedido que le efectuó PALAMEDES, sino que además, lo hizo con mucho agrado, ya que consideró sumamente modesta la recompensa solicitada,y ordenó a sus ayudantes que satisficieran de inmediato el pedido...(¡¡TODAVIA DEBEN ESTAR CONTANDO GRANOS!!).

Observando detenidamente el diagrama de flujo y el programa conexo, podemos notar que el comienzo del mismo está en la línea 130, donde se inicaliza la variable de control en 2. Esta variable también cumple, como es bastante frecuente, funciones operativas. Cabe destacar que hemos respetado estrictamente los principios que enunciamos en el punto 3 de esta misma nota.

FIGURA 4: FOR... / NEXT

```
100 FOR contador=valor inicial TO v. final STEP incremento
...      (  contenido del ciclo   )
200 NEXT contador
```

FIGURA 5: INDENTADO

```
100 FOR c = vi TO vf STEP s
110   LET x = x + s * c
120    PRINT x
130 NEXT C
```

LOS CICLOS FOR/NEXT

Es evidente que la sentencia BASIC más utilizada para ejecutar tareas repetitivas es la llamada FOR/NEXT. Esta sentencia compuesta (ocupa dos líneas del programa) tiene una estructura funcional que responde al tipo REPEAT ... UNTIL ..., dado que siempre se ejecuta -al menos una vez- el contenido del ciclo o por lo menos la primera instrucción interior, antes de verificarse el estado del contador.

La sintaxis completa de un ciclo FOR/NEXT es la de la Figura 4.

La familiaridad de esta sentencia nos exime de mayores comentarios sobre su sintaxis, pero solo mencionaremos que gran parte de la potencia de uso que posee puede atribuírsele a la posibilidad que brinda de usar VARIABLES como datos inicial, final e incremento de la VARIABLE DE CONTROL denominada CONTADOR, cuyos valores pueden fijarse en el momento de su ejecución, a través de algún INPUT o sentencia equivalente.

100 FOR contador = I TO F STEP S

Lo que sí deseamos tratar, aunque más no sea brevemente, es la APLICABILIDAD de los ciclos FOR/NEXT, es decir, los casos en los cuales se puede/conviene emplearlos y aquellos en que no es adecuado su uso.

Los casos en que es factible el empleo del FOR/NEXT son los que comprenden a un conjunto de instrucciones que deben repetirse UN NUMERO DETERMINADO DE VECES, es decir aquellos de los cuales conocemos el valor inicial y final del conteo que se produce al ejecutarlo y el valor del incremento del contador, y solo en estos casos.

Una condición extra del FOR/NEXT es relativa al incremento que se fija por medio del STEP, que debe ser CONSTAN-

FIGURA 6: LISTADO

```
100 REM *************************
105 REM *                       *
110 REM *  A L M A N A Q U E    *
115 REM *                       *
120 REM *************************
125 REM
130 REM * ACONDICIONAMIENTO SCREEN *
135 REM   ------------------------
140 SCREEN 0
145 WIDTH 40
150 KEY OFF
155 COLOR 4,14,14
160 DEF FN GCHAR(C,F) = VPEEK (F*40+C
    + INT ((40-PEEK (&HF3B0))/2+0.5))
165 REM * ASINACIONES M E S E S *
170 REM   --------------------
175 DIM MES$(12) , QD(12)
180 FOR I = 1 TO 12
185   READ MES$(I) , QD(I)
190 NEXT I
195 DATA Enero,31,Febrero,28,Marzo,31
    ,Abril,30,Mayo,31,Junio,30,Julio,
    31,Agosto,31,Setiembre,30,Octubre
    ,31,Noviembre,30,Diciembre,31
200 REM - - - - - - - - - - - - - - - -
205 REM * INGRESO DE DATOS *
210 REM   ----------------
215 CLS
220 LOCATE 0 , 1 : PRINT "AÑO? .....
    19__"
225 LOCATE 13 , 1 : LINE INPUT A$
230 LET AA = VAL(A$)
235 IF AA > 99 OR AA < 0 THEN GO TO
    225
240 LOCATE 0 , 3 : PRINT "MES?" :
    PRINT
245 FOR I = 1 TO 12
250  PRINT TAB(2) I ; TAB(6) "..... "
    ; MES$(I)
255 NEXT I
260 LOCATE 0 , 5 : LINE INPUT MX$
265 LET MM = VAL (MID$ (MX$ , 4 , 2))
270 IF MM < 1 OR MM > 12 THEN GO TO
    260
275 REM  * CALCULO AÑOS BISIESTOS *
280 REM    ----------------------
285 IF AA MOD 4 = 0  THEN LET BIS = 1
    ELSE LET BIS = 0
290 REM * CALCULO PASADO & FUTURO *
295 REM   ----------------------
300 LET PAS = 0 : LET FUT = 0.
305 LET M = 1
310 IF M > MM - 1 THEN GO TO 335
315  LET PAS = PAS + QD(M)
320  IF M = 2 AND BIS = 1 THEN LET
     PAS = PAS + 1
325  LET M = M + 1
330 GO TO 310
335 LET ND = QD(MM)
340 IF MM = 2 AND BIS = 1 THEN LET
    ND = 29
345 LET FUT = (365 + BIS) - (PAS + ND)
350 REM * CALCULO PRIMERO DE MES *
355 REM   ---------------------
360 LET D = (AA-1) * 365 + INT((AA-1)
    / 4) + PAS
365 LET PD = D - 7 * INT(D/7) + 3
370 IF PD = 9 THEN LET PD = 2
375 REM  * DISPLAY ALMANAQUE *
380 REM    -----------------
385 CLS
390 LOCATE 1 , 0 , 0 : PRINT STRING$
    (36 , 204 ) ; CHR$ (204)
395 FOR F = 1 TO 19
400  LOCATE 1 , F , 0 : PRINT CHR$(20
     3) ; TAB (37) ; CHR$ (203)
405 NEXT F
410 LOCATE 11 , 2 , 0 : PRINT MES$(MM
    ) ; " 19" + MID$ (STR$ (AA),2,2)
415 LOCATE 2 , 3 , 0 : PRINT PAS ;
    TAB(30) ; FUT
420 LOCATE 6 , 5 , 0 : PRINT "┌──────
    ──────────────────┐"
425 LOCATE 6 , 6 , 0 : PRINT "| Do Lu
    Ma Mi Ju Vi Sa |"
430 LOCATE 6 , 7 , 0 : PRINT "├──────
    ──────────────────┤"
435 LOCATE 6 , 8 , 0 : PRINT "|";
440 IF PD = 8 THEN LET COL = 1 ELSE
    LET COL = PD
445 FOR DIA = 1 TO ND
450  LET DIA$ = RIGHT$ (STR$ (DIA),2)
455  IF COL = 8 THEN LET COL = 1 :
     LOCATE 29 , , 0 : PRINT "|" :
     LOCATE 6 , , 0 : PRINT "|" ; TAB
     (29) " " : LOCATE 6 , , 0 :
     PRINT "|" ;
460  LOCATE ((COL-1) * 3 + 8) , , 0 :
     PRINT DIA$ ;
465  LET COL = COL + 1
470 NEXT DIA
475 LOCATE 29 : PRINT "|" : LOCATE 6
    , , 0 : PRINT "└──────────────
    ──────┘"
480 LOCATE 1 , 20 , 0 : PRINT STRING$
    (36 , 204 ) ; CHR$(204)
485 REM * R E C I C L A J E *
490 REM   -----------------
495 LOCATE 2 , 22 , 0 : PRINT "Salida
    por impresora ? (P/N)" ;
500 LET R1$ = INKEY$ : IF R1$ = ""
    THEN GO TO 490
505 IF R1$ = "P" OR R1$ = "p" THEN GO
    SUB 545
510 LOCATE 2 , 24 , 0 : PRINT "Otro
    mes ? (S/N)" ;
515 LET R2$ = INKEY$ : IF R2$ = ""
    THEN GO TO 505
520 IF R2$ = "s" OR R2$ = "S" THEN
    CLS : GOTO 240
525 CLS
530 KEY ON
535 COLOR 1 , 3 , 3
540 END
545 REM * SALIDA IMPRESA *
550 REM   --------------
555 LPRINT STRING$ (64 , 95) ;
560 LPRINT CHR$ (10)
565 FOR F = 2 TO 17
570   FOR C = 2 TO 34
575     LPRINT CHR$ (FN GCHAR (C,F));
580     LPRINT CHR$ (32) ;
585   NEXT C
590   LPRINT CHR$ (10) ;
595 NEXT F
600 LPRINT STRING$ (64 , 95)
605 RETURN
```

TE, eso no obsta a que su valor se establezca a través de las denominadas "variables", pero ese incremento, una vez determinado, debe mantenerse fijo durante todo el ciclo. En el caso poco frecuente de que se requiera dar a la variable de control un carácter operativo y que se necesite variar el incremento en cada "vuelta" del ciclo, deberemos recurrir a otro tipo de sentencia más flexible.
A efectos de mejorar la LEGIBILIDAD del FOR/NEXT aconsejamos indentar (hacer una sangría) el interior del ciclo, tal como se muestra en la Figura 5.
Otros factores a tener en cuenta con estos ciclos son: a) no conviene verificar el valor de contador dentro del ciclo, y b) no es aconsejable modificar el valor del contador dentro del ciclo.

OBSERVACIONES FINALES

Los ciclos REPEAT ... UNTIL ... se pueden utilizar en todos aquellos casos en que no deba verificarse PREVIA Y NECESARIAMENTE la condición de salida



del ciclo, y en los que - obviamente - se conozcan los valores inicial y final del contador.
Para finalizar hemos agregado el listado de un programa que preparó un grupo de alumnos adolescentes y que fue denominado "ALMANAQUE". En dicho programa se hace uso frecuente de estructuras del tipo ciclo REPEAT, salvo en el bloque de Cálculo de los Días Pasados y Futuros, en el cual el ciclo es del tipo WHILE ... DO ...
Quienes no poseen impresora pueden abstenerse de copiar las líneas 495-505 y la subrutina correspondiente, o mejor aún, copiarlas, grabar el programa en casete o disquete y animarse a acercarse al CENTRO DE ASISTENCIA AL USUARIO DE TALENT MSX más cercano a su domicilio para sacar una copia impresa de su hoja. (No olvidar las masas para la secretaria...).

NOTA: El original de este artículo fue confeccionado por el autor con un equipo SEMI-PROFESIONAL de la línea MSX, compuesto por una microcomputadora hogareña TALENT MSX DPC-200, Unidad de Discos Flexibles TALENT DPF-550 de 360 Kb, pantalla de TV color e Impresora SEIKOSHA SP-1000AS de 100 cps y tipografía NLQ. Como Procesador de Textos se ha utilizado un MSX-Write en cartucho de ASCII Corp traducido al castellano y producido localmente por TELEMATICA S.A.

Gustavo O. Delfino

CONCURSO
EL PROGRAMADOR DEL AÑO '88 K64

BASES

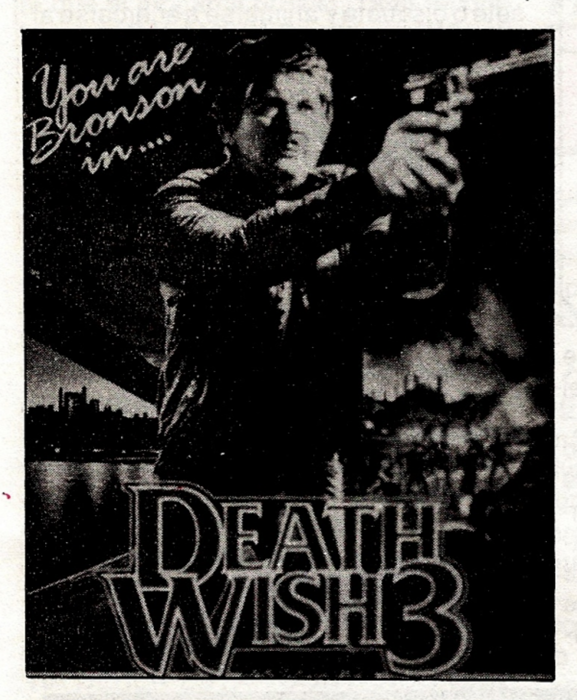
Una vez terminado y revisado tu programa, deberás enviarlo a la editorial grabado en un cassette o diskette, varias veces para mayor seguridad. (Inclusive grabado con dos grabadores distintos). Indicar en el cassette o diskette, los datos del programa, computadora y autor.
Otra condición es que sea original e inédito, es decir que no haya sido enviado a ninguna otra publicación.
Si bien es preferible que vaya acompañado del listado del mismo por impresora, éste no es imprescindible.
El programa deberá venir con un texto que aclare cuál es su nombre, objetivo, modo de uso, y explicación de cada una de sus partes, subrutinas y variables. Si posee lenguaje de máquina, es fundamental una buena explicación sobre su funcionamiento e ingreso a la máquina. No olvidarse los datos completos del autor o autores.
El texto se presentará en hojas tipo oficio y mecanografiado a doble espacio. No importa que la redacción no sea muy clara, eso queda por nuestra cuenta.

IMPORTANTES PREMIOS

Los daremos a conocer próximamente.

CIERRE: **El cierre de recepción de trabajos para concurso de programas será el 30/9/88 . (K64 se reserva el derecho de publicación de los programas recibidos, como asimismo la devolución del material).**

DEATH WISH 3

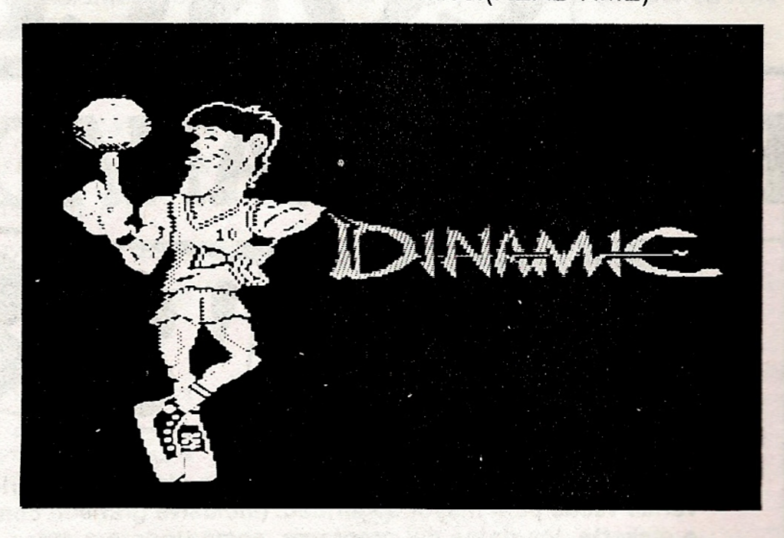


Los robos y los asaltos en la ciudad se han convertido en habituales y parece que la policía no puede hacer nada. Es entonces que aparece Paul Kersey, el Vengador Anónimo, quien debe liberar a Nueva York de todos los criminales que pululan por sus calles.
La labor no es fácil, y muchas veces los transeúntes inocentes resultarán heridos. Los policías ayudan en lo que pueden, pero atención: si se les dispara a alguno de ellos, verán que no son tan amistosos.
DEATH WISH 3 tiene un escenario tridimensional. Se puede girar para cualquier dirección y la vista que se obtiene siempre será lógica geográficamente.
Una brújula en la parte inferior de la pantalla, junto a un radar y un listado de la policía, son los elementos de los que disponemos como ayuda.
Luego está nuestro arsenal. Hay cuatro tipo de armas a nuestra disposición: el famoso Magnum 475, una escopeta de cañones recortados, una metralleta y un mortero. Cada arma tiene un efecto distinto pero devastador, y la munición no es ilimitada.
Se puede entrar en los edificios para buscar a los jefes mafiosos, o armas adicionales. Tambien podemos acercarnos a las ventanas y actuar como francotiradores.
Las teclas de control son las siguientes: izquierda (Z), derecha (X), girar arriba (O), girar abajo (K), disparo (ESPACIO), para entrar por las puertas (RETURN), para mirar por las ventanas (W), para cambiar arma (SELECT), para cambiar mapa (armas o jefes) (M), pausa (STOP) y abandonar el juego (CTRL y ESC). (MICROBYTE)

KNIGHT COMMANDER

La primera cosa que se puede decir de este programa es que añade 40 nuevos comandos al BASIC del MSX, dejando la memoria RAM completamente libre.
El KNIGHT COMMANDER permite al usuario comprimir los programas en BASIC borrando espacios innecesarios y uniendo tantas líneas como sea posible. Esto ahorra memoria y hace que el programa sea más rápido.
También incluye funciones de búsqueda, reloj de tiempo real, indicador parpadeante de cursor, volcado de la pantalla a la impresora, almacenamiento y recuperación de pantallas en la memoria, recuperación de un programa borrado con NEW, volcado de variables a la impresora, mejoras en la ejecución de un programa paso a paso, almacenamiento de la pantalla gráfica, almacenamiento de gráficos en casete y 22 teclas con comandos BASIC.
Junto con KNIGHT COMMANDER viene un juego para la demostración de los nuevos comandos.
Dick Sham, el creador de este utilitario, permite a todos los programadores beneficiarse con los comandos BASIC suplementarios, manteniendo toda la memoria libre. Los programadores expertos valorarán el hecho de que el KNIGHT COMMANDER haya sido creado usando los programas KNIGHTS EXPERT MACHINE CODE y el KNEASE-KNIGHTS EASY ASSEMBLER. (MICROBYTE)

FERNANDO MARTIN BASKET MASTER

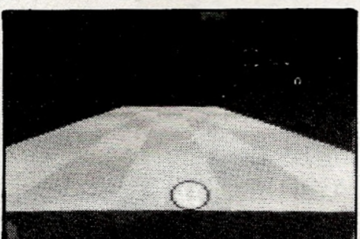


Entre los video juegos deportivos éste es uno de los más famosos y publicitados del mercado español.
En él debemos enfrentar en un partido de basket al internacional Fernando Martín. Pueden participar uno o dos jugadores, en tres diferentes niveles de juego.
Como en todo partido de basket, la posesión del balón es fundamental. Si el jugador no lleva el balón, la posición del cuerpo será la de la dirección de avance. En cambio, cuando lo lleva no cambiará la posición del cuerpo, pero se desplazará en la dirección elegida, lo que permitirá cubrir el balón del contrincante.
El programa permite tiros al aro desde cualquier posición e incluso entrar en bandeja. Podemos ser brillantes "reboteros" o tener un dribbling infernal.
La estrategia cuenta, y mucho. A medida que juguemos nos iremos dando cuenta de cuáles son los puntos débiles del rival y cuáles nuestras mayores virtudes. Obviamente deberemos tratar de llevar el juego hacia esas características.
En cuanto a las faltas, son las normales en un juego de basket, y la computadora nos sancionará si las cometemos.
El partido consta de dos tiempos de cinco minutos. También puede finalizar si alguno de los dos jugadores supera el límite de cinco faltas personales.(REAL TIME)

TRAILBLAZER

Una opción diferente. En este juego, de origen brasilero, debemos conducir una pelota de fútbol a través de unos corredores que toman formas de lo más variadas.
Existen catorce pistas diferentes, cada una con su nombre. Algunas de ellas no pueden ocultar su origen: "Bem fácil", "O saltador", "Pula pula", "Av. Brasil", "O as do salto", "Corajoso", "Corra e pu-





le","Bolinha de ouro", "Nao desanime", etcétera.

Algunas son fáciles y otras más difíciles, pero eso no es tan importante debido a que se establecen y quedan registrados los mejores tiempos en cada una de las pistas.

También se puede jugar por puntos. Aquí debemos cumplimentar cada etapa en un tiempo máximo para no quedar eliminados.

Pulsando la letra M podremos sacar o poner la música de fondo, muy pegadiza por cierto. (REAL TIME)

DUSTIN

Dustin ha caído en prisión y debe sobrevivir y buscar la forma de escapar de la misma.

Los guardias tienen cigarillos, y Dustin los quiere. Como no tiene dinero para comprárselos la solución es golpearlos y hacer uso de su fuerza bruta. Luego irá a ver a los otros prisioneros para canjear esos elementos por relojes o encendedores y así seguirá buscando elementos que le permitan pasarla bien y concretar la tan ansiada fuga.

No hay que impacientarse ya que el juego lleva su tiempo. Prestemos atención y ayudemos a Dustin en su "misión". (MICROBYTE)



SPIRITS

En esta aventura nos transformaremos en magos.

Estamos encerrados en dos castillos comunicados entre si por pasillos con puertas secretas. Nuestra misión es la de rescatar a una princesa y un Caballero secuestrados por un brujo y custodiados por el Aguila Infernal.

A lo largo de los pasillos encantados de los castillos, nos interceptarán fantasmas, gatos y un arquero que se empecina en afinar su puntería utilizándonos de blanco.

Cualquiera de estos enemigos que nos toque, consumirá nuestra energía.

El juego se acaba cuando se nos agote la tercera carga de energía.

Para defendernos de nuestros verdugos, la única arma que tenemos, son nuestros poderes mágicos. Lanzando hechizos, los fantasmas y demás enemigos se paralizan.

Esta es nuestra oportunidad de avanzar en nuestro objetivo. La parálisis no dura demasiado, depende de la habilidad del jugador, el provecho que le saque a esos escasos minutos.

La pantalla se encuentra dividida en dos partes. La superior va mostrando cómo se desarrolla la aventura mientras que en la parte inferior, tiene lugar el seguimiento de los objetos y los personajes que el mago tiene que encontrar para cumplir su misión.



FIGURA 1

```
10 REM SPIRITS
20 COLOR 1,1,1:SCREEN 2,2
30 BLOAD"CAS:",R
40 BLOAD"CAS:",R
50 BLOAD"CAS:",R
60 BLOAD"CAS:",R
70 BLOAD"CAS:",R
80 FOR I=44450! TO 44459!:READ A
:POKE I,A:NEXT:POKE &HD14,187:PO
KE &HAD15,64
90 DEFUSR=44244!:A=USR(0)
100 DATA &HAF,&H32,46,201,&H32,1
34,202,&HC3,&H83,&HBB
```

El primer objeto a buscar, es la bola de cristal para poder averiguar dónde está la vara del poder y el águila gigante que tenemos que destruir. Esta vara es esencial para desencantar la armadura hechizada.



Cuando aún no tenemos la esfera, ésta aparecerá en la zona inferior, dandonos una pista para encontrarla.

Sin la vara del poder, el mago no tiene demasiados recursos para defenderse. Puede moverse hacia la derecha, izquierda, subir o bajar las escaleras y lanzar rayos paralizantes. En cambio cuando usemos la vara, podemos accionar palancas que lanzan rastrillos o activar trampas.

El juego se detiene cuando pulsemos "H".

La calidad de los personajes: fantasmas, jorobados, hombre lobo, arqueros, como así también la escenografía, merecen ser destacada.

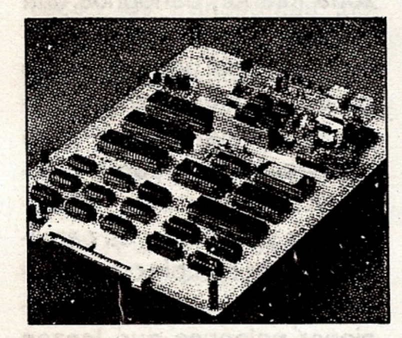
Es un juego que atrapará al participante debido al desafío que propone cada pasillo. En la figura 1 tenemos un listado cargador del juego que nos proporciponará vidas infinitas. (REAL TIME)

Para comunicarse con nosotros deben escribirnos a "Load MSX", Paraná 720 ,5º Piso, (1017), Capital Federal.

AFICIONADO DX

Soy aficionado al DX (diexismo , "cazador de ondas cortas") .
Si alguien está interesado en este tema con relación a la computación, se puede poner en contacto conmigo.
Respecto a la nota sobre MSX-FAX, mencionan que su manejo es igual al de un cartucho de comunicaciones, ¿de cuál se trata, dónde se consigue y cómo funciona ?
El manual de usuario de mi SEIKOSHA GP-550A está escrito en inglés y desearía conocer más a fondo las posibilidades de dicha impresora, ¿alguien puede darme una mano?

Mario Enrique Muñoz
Vicente Lopez y Planes 2268
7600-Mar del Plata-RA



Load MSX

Tal como se solicita publicamos la dirección, para posibilitar contactos entre aficionados a determinados hobbys en común.
La nota sobre el cartucho MSX-FAX contesta por sí sola estas dudas; aclarando la misma, diremos que las comunicaciones referidas son las llamadas "facsímiles".
Estos son documentos escritos en papel y luego transmitidos a través de la red telefónica para ser recibidos en cualquier otra terminal de facsímil.
También se menciona que dicho cartucho esta aún en desarrollo, ¡paciencia amigo!

USUARIA EDUCATIVA

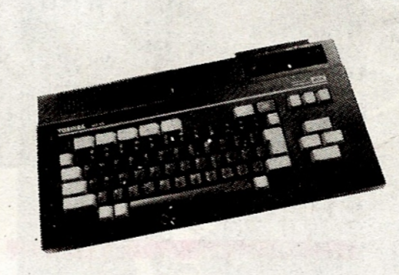
Soy estudiante de Análisis de sistemas en la UTN de Rosario y estoy dando clases en el colegio secundario de mi pueblo .
Mi problema se presenta en la parte gráfica: no sé cómo dibujar contornos de figuras a modo de un lápiz sobre el papel.
En el Nro. 17 se menciona la posibilidad de obtener programas educativos sin cargo en forma periódica; he cumplido todos los requisitos mencionados pero no obtuve respuesta.

Stella Maris Santandrea
Santa Fe

Load MSX

Con relación a la primera consulta , la solución consiste en adicionar a cualquiera de las consolas un paquete de manejo de gráficos por lápiz óptico, que permite realizar dibujos sobre la pantalla del televisor como si fuera un tablero de dibujo.
La segunda alternativa es más económica pero más laboriosa ya que es necesario efectuar un diseño previo sobre papel cuadriculado para luego volcarlo, mediante algún programa de soporte, en la RAM de video y almacenarlo como pantalla en disco.
Las cartas dirigidas a Telemática S.A. con el objeto de obtener los paquetes educativos deben enviarse con algún distintivo oficial (papel y/ o sobre) que identifique al establecimiento de enseñanza en particular, además de la/s factura/s de compra mencionada/s.

PROBLEMAS ACENTUADOS I



Poseo una Toshiba HX20 junto con una impresora Texas Instruments Omni 800 / Modelo 855 y me he encontrado con que algunos programas que debieran funcionar en una Talent DPC 200 no lo hacen en mi consola como es el caso del programa "Acento" del Nro. 19 .
En principio funciona para pruebas aisladas de impresión , ¿qué estoy haciendo mal?

Mario Raúl López
Entre Ríos

PROBLEMAS ACENTUADOS II

Quisiera solicitarles información sobre el programa "Acento" para impresoras no MSX .
Una vez cargado , si presiono los caracteres de las vocales acentuadas del teclado solo se imprimen vocales.
La impresora que poseo es una Compuprint modelo X-80.

María Cristina Roca Azul

Load MSX

No es norma de la redacción revisar listados enviados , pero a título de curiosidad se ha releído la impresión enviada por Mario , encontrándose un error bastante habitual y, por lo tanto, se considera digno de mención ya que puede ser útil en este caso así como en sucesivas ocurrencias del mismo.
En la línea 10130 del listado número 1, sobre el final del renglón, dice CB (CE-BE) en lugar de C8 (CE -OCHO). Se recomienda , en general , prestar atención a las ocurrencias de dichos caracteres en los listados ya que un programa escrito en lenguaje ensamblador no perdona ningún cambio.
Otro punto a tener en cuenta es la correcta secuencia de caracteres que el programa envía hacia la impresora para "simular" los caracteres en cuestión. Es decir, deben verificarse los códigos de retroceso de carro así como los correspondientes al acento y al símbolo " ñ" según se desprende del listado.
2 . Lamentablemente no contamos con el manual de la impresora mencionada por María Cristina para verificarlo.



AÑO 3 Nº 36 ≠ 12 REP. ARGENTINA

COMPUTACION



PARA TODOS



ESTUDIANTES

PARA QUE SIRVE LA COMPUTADORA

NUEVO CONCURSO MENSUAL

PROFESIONALES

SOFT PARA MEDICOS Y CONTADORES

LOS ERRORES DE LA PROGRAMACION

SPECTRUM - CZ - TK - DREAN COMMODORE - TI - MSX - ATARI y PC

Encienda una computadora Talent MSX y sus periféricos.

MODEM
- Interfaz asincrónica incorporada.
- Normas BELL y CCITT, 300 y 1.200 baudios.
- Incluye en 80 Kb software de comunicaciones, MSX-PLAN y MSX-WRITE.

UNIDAD DE DISCOS FLEXIBLES
- 5 1/4" de 360 Kb (DS-DD)
- Velocidad transferencia 250 Kb/sg.
- Formato grabacion compatible MS-DOS.

MINI-LAN
EQUIPO CENTRAL PROFESOR
IMPRESORA
EQUIPO ALUMNO 1
2
3
4
5
6
7
8
9
10 EQUIPO ALUMNO
DISKETTERAS 2 x 360 KB

(RED PARA USO EDUCACIONAL)
- Comunicación por línea compartida a 30.000 baudios.
- Comparte unidades de disco e impresoras de consola MSX maestro con hasta 10 consolas MSX alumno
- Carga simultánea de un programa a todos los alumnos.
- Carga individual de maestro a alumno
- Salvado de programa alumno a unidad de disco maestro.
- Salida a impresora de maestro del listado de programa alumno, en spooling.
- Funciona desde MSX-Basic, MSX-Logo o cualquier programa que corra desde Basic.
- Estando activa, se dispone de todo el MSX-Basic.

RS-232
- Velocidades programables desde 50 a 19.200 baudios.
- Procesamiento de comunicaciones desde Basic.
- Buffer de 128 caracteres para recepción

TECLADO NUMERICO
- Conexion a Joystick
- Se integra a todo el software que corre bajo MSX-DOS (Ej.: D-Base II, MSX-Plan, etc.)

MOUSE
- Código DPM-220, accesorio para graficar.

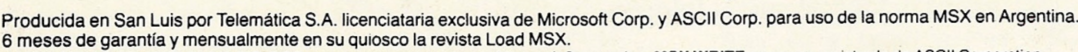
MONITOR MONOCROMATICO 12"
- Anti-reflejo - Fósforo verde.
- Apto para uso profesional.
- (80 caracteres x 25 líneas).
- Parlante con amplificador incorporado.

EXPANSION 80 COLUMNAS
- Hace posible la utilización de software estandar CP/M, emulando terminal tipo VT-52.
- Incluye software para manejo de video.

Software

MSX-LOGO	MSX-LPC	MSX-PLAN	MSX-WRITE
Desarrollado por Logo Computer System Inc. con aplicación de primitivas y redacción del Manual por los Ings. Hilario Fernández Long y Horacio Reggini.	Lenguaje de programación estructurado y en castellano.	Planilla de calculo de Microsoft Corp. (Versión para MSX del Multiplan.)	Procesador de palabra de ASCII Corp. en castellano.




Producida en San Luis por Telemática S.A. licenciataria exclusiva de Microsoft Corp. y ASCII Corp. para uso de la norma MSX en Argentina.
6 meses de garantía y mensualmente en su quiosco la revista Load MSX.

- **MSX, MSX-DOS, MSX-PLAN, MS-DOS**, son marcas registradas de Microsoft Corporation. **MSX WRITE** es marca registrada de ASCII Corporation.
- **CP/M** es marca registrada de Digital Research. **MSX-LOGO** es marca registrada de Logo Computer Systems Inc. **Telemática:** 1986. Todos los derechos reservados.

Los datos y especificaciones que figuran en este aviso pueden ser modificados sin previo aviso.